

**SISTEM *SWITCHING* DAN *MONITORING* TEGANGAN DAN ARUS
PADA *SMART GRID* BERBASIS RASPBERRY PI 3 TERINTEGRASI
DENGAN MEDIA SOSIAL DAN *WEBSITE***

SKRIPSI

TEKNIK ELEKTRO KONSENTRASI TEKNIK ELEKTRONIKA

Diajukan untuk memenuhi persyaratan
memperoleh gelar Sarjana Teknik



**OWLENA RENASEILLA YOUSI
NIM. 145060301111001**

**UNIVERSITAS BRAWIJAYA
FAKULTAS TEKNIK
MALANG**

2018

LEMBAR PENGESAHAN
SISTEM *SWITCHING* DAN *MONITORING* TEGANGAN DAN ARUS
PADA *SMART GRID* BERBASIS RASPBERRY PI 3 TERINTEGRASI
DENGAN MEDIA SOSIAL DAN *WEBSITE*
SKRIPSI

TEKNIK ELEKTRO KONSENTRASI TEKNIK ELEKTRONIKA

Ditujukan untuk memenuhi persyaratan
memperoleh gelar Sarjana Teknik



OWLENA RENASEILLA YOUSI
NIM.145060301111001

Skripsi ini telah direvisi dan disetujui oleh dosen pembimbing
pada tanggal 19 Januari 2018

Dosen Pembimbing I

Adharul Muttaqin, S.T., M.T.
NIP. 19760121 200501 1 001

Dosen Pembimbing II

Eka Maulana, S.T., M.T., M.Eng.
NIP. 201201 841130 1 001

Mengetahui,

Ketua Jurusan

Ir. Hadi Suyono, S.T., M.T., Ph.D., IPM.
NIP. 19730520 200801 1 013

JUDUL SKRIPSI:

SISTEM SWITCHING DAN MONITORING TEGANGAN DAN ARUS PADA SMART GRID BERBASIS RASPBERRY PI 3 TERINTEGRASI DENGAN MEDIA SOSIAL DAN WEBSITE

Nama Mahasiswa : OWLENA RENASEILLA YOUSI

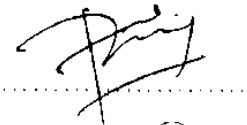
NIM : 145060301111001

Program Studi : TEKNIK ELEKTRO

Konsentrasi : TEKNIK ELEKTRONIKA

Komisi Pembimbing :

Ketua : Adharul Muttaqin, S.T., M.T.

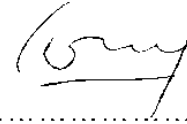


Anggota : Eka Maulana, S.T., M.T., M.Eng.

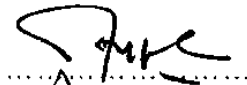


Tim Dosen Penguji :

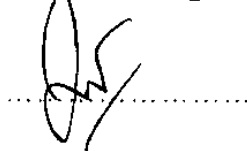
Dosen Penguji 1 : Dr. Ir. Ponco Siwindarto, M.Eng.Sc.



Dosen Penguji 2 : Ir. Nurussa'adah, M.T.



Dosen Penguji 3 : Akhmad Zainuri, S.T., M.T.



Tanggal Ujian : 12 Januari 2018

SK Penguji : 020/UN10.F07/SK/2018

*Teriring Ucapan Terima Kasih kepada:
Ayahanda dan Ibunda tercinta*

PERNYATAAN ORISINALITAS SKRIPSI

Saya menyatakan dengan sebenar-benarnya bahwa sepanjang pengetahuan saya dan berdasarkan hasil penelusuran berbagai karya ilmiah, gagasan dan masalah ilmiah yang diteliti dan diulas di dalam Naskah Skripsi ini adalah asli dari pemikiran saya. Tidak terdapat karya ilmiah yang pernah diajukan oleh orang lain untuk memperoleh gelar akademik di suatu Perguruan Tinggi, dan tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain, kecuali yang secara tertulis dikutip dalam naskah ini dan disebutkan dalam sumber kutipan dan daftar pustaka.

Apabila ternyata di dalam naskah Skripsi ini dapat dibuktikan terdapat unsur-unsur jiplakan, saya bersedia Skripsi dibatalkan, serta diproses sesuai dengan peraturan perundang-undangan yang berlaku (UU No. 20 Tahun 2003, pasal 25 ayat 2 dan pasal 70).

Malang, 19 Januari 2018

Mahasiswa,



Owlana Renaseilla Yousi

NIM. 145060301111001

RINGKASAN

Owlana Renaseilla Yousi, Jurusan Teknik Elektro Fakultas Teknik Universitas Brawijaya, 2 Januari 2018, *Sistem Switching dan Monitoring Tegangan dan Arus pada Smart Grid Berbasis Raspberry Pi 3 Terintegrasi dengan Media Sosial dan Website*: Adharul Muttaqin, Eka Maulana.

Smart grid merupakan jaringan listrik yang menggunakan teknologi digital untuk memantau atau mengelola transportasi listrik dari beberapa sumber. *Smart Grid* akan terus mengalami kemajuan seiring perkembangan teknologi. Dalam *smart grid* diperlukan sistem *monitoring* dan proses *switching* dengan tepat untuk kepentingan keamanan sistem agar sistem dapat berjalan dengan baik.

Dalam penelitian ini, dirancang sebuah sistem yang mampu melakukan proses *switching* dan pengiriman data pada sistem *monitoring smart grid* secara *wireless* dan dapat diakses pada laman website, Google Drive dan media sosial yaitu Telegram. Dengan demikian, *monitoring* dan proses *switching smart grid* lebih mudah diakses dan dapat dikomunikasikan secara jarak jauh. Sistem tersebut membutuhkan sebuah web *server* yang dijalankan oleh Raspberry Pi 3 yang berfungsi untuk menerima data dari modul sensor yang terpasang pada masing-masing *node* dan akan disimpan pada *database* MySQL.

Berdasarkan hasil penelitian, data *monitoring* dikirim dalam bentuk file *.csv mempunyai ukuran yang kecil, yaitu file yang berisi 1000 baris data dengan ukuran 35,5 KB. Pemrograman web menggunakan PHP yang berfungsi untuk menampilkan data dari *database* MySQL dalam bentuk grafik garis yang memanfaatkan *library* Highcharts. Grafik yang menampilkan informasi nilai tegangan dan arus pada tiga *node* dapat dibuka sekaligus melalui *browser* dengan menggunakan *ip address* dari *server* Raspberry Pi 3. Data *monitoring* dapat diakses di manapun dan kapanpun dalam Google Drive. Pengiriman 500 baris data dengan ukuran data 18 KB memerlukan waktu 2,027 detik. Untuk mengantisipasi salah satu *user* yang mengalami kendala komunikasi, maka digunakan dua *user* yang beroperasi. Penggunaan dua *user* secara bersamaan menyebabkan *user* 2 lebih lambat menerima data dibandingkan dengan *user* 1, dengan selisih waktu pengiriman 0,56 detik. Proses *switching* pada *smart grid* bertujuan untuk menyambungkan dan memutuskan aliran listrik dari sumber untuk kepentingan penanganan yang cepat jika terjadi masalah yang tidak diinginkan, proses *switching* bekerja melalui pesan tertentu yang dikirimkan Telegram, pesan tersebut akan diproses oleh program Python sehingga dapat mengirimkan perintah *ON* atau *OFF* untuk *node* yang dituju.

Kata kunci: *Smart grid*, *Monitoring*, *Switching*, *Server* Raspberry Pi 3, Python, media sosial Telegram, *Website*, *database* MySQL

SUMMARY

Owlena Renaseilla Yousi, *Department of Electrical Engineering of University of Brawijaya, 2 January 2018, Switching System and Monitoring Voltage and Current for Smart Grid Based Raspberry Pi 3 Integrated with Social Media and Website: Adharul Muttaqin, Eka Maulana.*

Smart Grid is an electrical network that uses digital technology to monitor or manage electrical transport from multiple sources. Smart Grid will continue to progress with the development of technology. The Smart Grid monitoring system and the process switching is required for system protection so that the system works correctly

In this research, a system is designed that is able to perform the process of switching and sending data on the smart grid system wirelessly and can be accessible on the web page, Google Drive, and social media that is Telegram. Therefore, monitoring and switching smart grid more accessible and can be communicated remotely. Smart grid monitoring and switching systems require a web server run by Raspberry Pi 3 that serves to receive data from the sensor module installed on each node and will be stored in the MySQL database.

*Based on the results of the research, data sent with *.csv format type file has a small size, that is a file containing 1000 rows of 35.5 KB data sizes. Web programming using PHP that serves to display data from the MySQL database in the form of line graphs that utilize the library Highcharts. Graphs that display voltage and current values information on three nodes can be opened at once through a browser using ip address from Raspberry Pi 3 server. Monitoring data can be accessed anywhere within Google Drive. Sending 500 data lines with a data size of 18 kB takes 2,027 seconds. To anticipate one user who experiences communication problems, two users are used. The use of two users ensures that user 2 receives slower data compared to user 1, with a difference of 0.56 seconds. The switching process on the smart grid aims to connect and disconnect the current from sources for quick processing in case of unwanted problems, the switching process works via a specific message sent by Telegram, the message will be processed by the Python program so that it can send commands ON or OFF for the intended node.*

Keywords: *Smart Grid, Monitoring, Switching, Server Raspberry Pi 3, Python, Telegram social media, Website, MySQL database*

PENGANTAR

Alhamdulillah, puji dan syukur penulis panjatkan kehadiran Allah SWT, karena atas segala petunjuk dan nikmat-Nya lah skripsi ini dapat diselesaikan. Sholawat serta salam tidak lupa penulis haturkan kepada junjungan Rasulullah Muhammad SAW, semoga kelak mendapatkan syafaat beliau di yaumul qiyamah.

Skripsi berjudul “Sistem *Switching* dan *Monitoring* Tegangan dan Arus pada *Smart Grid* Berbasis Raspberry Pi 3 Terintegrasi dengan Media Sosial dan *Website*” ini disusun untuk memenuhi persyaratan memperoleh gelar Sarjana Teknik di Jurusan Teknik Elektro Universitas Brawijaya.

Penulis menyadari bahwa dalam penyusunan skripsi ini tidak terlepas dari bantuan berbagai pihak. Oleh karena itu, dengan ketulusan dan kerendahan hati penulis menyampaikan terima kasih kepada:

1. Kedua orang tua penulis, Bapak Hairullah dan Ibu R. Titik Susilowati atas segala doa, pengorbanan, inspirasi, nasehat, kasih sayang, perhatian dan kesabarannya didalam membesarkan dan mendidik penulis, serta adik-adik penulis, Denaya Mahabab Yousi dan Ayunya Kinar Yousi atas segala dukungan, semangat, dan motivasi hingga terselesaikannya skripsi ini.
2. Yang Terhormat Bapak M. Aziz Muslim, S.T., M.T., Ph.D. selaku Ketua Jurusan Teknik Elektro Universitas Brawijaya.
3. Yang Terhormat Bapak Hadi Suyono, S.T., M.T., Ph.D. selaku Sekretaris Jurusan Teknik Elektro Universitas Brawijaya.
4. Bapak Ali Mustofa, S.T., M.T. selaku Ketua Program Studi Sarjana Teknik Elektro Universitas Brawijaya.
5. Ibu Ir. Nurussa'adah, M.T. selaku Ketua Kelompok Dosen Keahlian Elektronika Jurusan Teknik Elektro Universitas Brawijaya, Dosen Penasehat Akademik penulis atas segala bimbingan, pengarahan, saran, dan kritik yang telah diberikan selama perkuliahan dan pengerjaan skripsi.
6. Yang Terhormat Bapak Adharul Muttaqin, S.T., M.T. selaku Dosen Pembimbing I atas segala bimbingan, pengarahan, saran, dan kritik yang telah diberikan selama proses pengerjaan skripsi.

7. Yang Terhormat Bapak Eka Maulana, S.T., M.T., M.Eng. selaku Dosen Pembimbing II atas segala bimbingan, pengarahan, saran, dan kritik yang telah diberikan selama proses pengerjaan skripsi.
8. Bapak Akhmad Zainuri, S.T., M.T. selaku Dosen Penasehat Akademik penulis atas segala bimbingan, pengarahan, saran, dan kritik yang telah diberikan selama masa studi.
9. Seluruh dosen pengajar Jurusan Teknik Elektro Universitas Brawijaya.
10. Seluruh *staff recording* Jurusan Teknik Elektro Universitas Brawijaya.
11. Bapak Mulyadi, S.T. selaku Laboran Laboratorium Elektronika atas bantuan, fasilitas dan keramahannya selama penulis menjadi asisten laboratorium dan pengerjaan skripsi.
12. Moh. Ghadafi, S.T. yang telah memberi dukungan, semangat, motivasi dan doa selama ini, serta turut membantu proses pengerjaan skripsi.
13. Machfud Firmansyah, S.T., Alfian Khairi, Hasan, Febrian Daniel, Ikhwan Fajri, Hafidin, Rizky, Reza, Bill, Yunda, Gerdy Hasni yang turut membantu secara langsung dalam proses pengerjaan skripsi.
14. Keluarga Besar Mahasiswa Teknik Elektro Universitas Brawijaya.
15. Teman-teman DIODA 2014 atas segala bantuan dan kebersamaan yang telah diberikan selama masa studi.
16. Teman-teman Bidadari DIODA 2014, Muthia, Titah, Titi, Arum, Cinta, Dini, farra, Regi, Noska, Luthfiah, Dina, Nola, Lita, Pegy, Putri, Septi, Sipa, Vio, Winda, Zulva, Shinta atas segala dukungan dan kebersamaannya selama masa studi.
17. Teman-teman konsentrasi Teknik Elektronika Universitas Brawijaya.
18. Keluarga Besar Laboratorium Elektronika, Mbak Itsna, Mbak Novvy, Mbak Muhiroh, Mbak Hani, Mas Yana, Mas Dafi, Mas Ari, Mas Naufal, Mas Andy, Mas Adam, Mas Yuda, Fajri, Hafidin, Rizky, Febrian, Rivan, Gagah, Haekal, Dina, Putri, Annabel, Lina, Arya, Bagas, Yunda, Ade, Bill, James dkk atas kerjasamanya, segala pengalaman, kebersamaan dan bantuan selama menjadi asisten.
19. Keluarga Besar Divisi Otomasi, Mas Dimas, Mas Faisal, Mas Rozi, Mas Ikhfal, Mas Naufal, Mas Andri, Mas Dennis, Mas Rifan, Mas Fauzan, Mas Zaini, Mas Kevin, Mas Wicak, Mas Mukti, Mbak Diana, Mbak Keyko, Lita, Reynald, Anthony, Dammegi, Anwi, Yusril, Karil, Izzul, Gozal, Indra, Dani, Eka, Mukti, Angga, Ardi, Jet-jet, Ilham, Jason,

Najib, Agung atas pengalamannya mempelajari PLC dan berjuang mendapatkan piala dalam berbagai kompetisi PLC.

20. Teman-teman pejuang skripsi, Hasan, Febrian, Fajri, Putri, Rivan, Dina, Luthfiah dan Ivan, atas segala bantuan dan motivasi selama pengerjaan skripsi.
21. Teman-teman anggota Unik dan Kreatif, Doni, Hasan, Esti, Mas Machfud, Mas Ari, Mas Arsyil, Mas Fatah, Lita, Amrul, Mas Ilham, Mas Judin, Mbak Novvy, Mas Naufal dkk atas segala kerjasama dan pengalaman dalam menjalankan Tri Dharma Perguruan Tinggi Bersama dosen pendamping Bapak Eka Maulana, S.T., M.T., M.Eng.
22. Teman-teman PKL PT. Petro Kimia Gresik, Febrian, Dina, Putri, Kiki, Rizal atas kerjasama, motivasi, dan pengalaman kerja.
23. Sahabat-sahabat solehah yang setia, Rita, Puput, Nindy, Yeni, Billy, Naura, Miming.
24. Teman-teman Kost Putri Kertoleksono 19 A, Mbak Isti, Mbak Febry, Mbak Maria, Mbak Isma, Mbak Eva, Sani, Fety Gita, Hasna, Mita, Indah, Ayu, Azizah, Nafi Henny, Rina, Vikka atas kerjasama dan kebersamaan selama di kost.
25. Seluruh teman-teman serta semua pihak yang tidak mungkin bagi penulis untuk mencantumkan satu-persatu, terimakasih banyak atas bantuan dan dukungannya.

Pada akhirnya, penulis menyadari bahwa skripsi ini masih belum sempurna. Oleh karena itu, penulis sangat mengharapkan kritik dan saran yang membangun. Penulis berharap semoga skripsi ini dapat bermanfaat bagi pengembangan ilmu pengetahuan dan teknologi serta bagi masyarakat.

Malang, Januari 2018

Penulis

DAFTAR ISI

PENGANTAR	i
DAFTAR ISI	v
DAFTAR TABEL	ix
DAFTAR GAMBAR	xi
DAFTAR LAMPIRAN	xiii
BAB I PENDAHULUAN	1
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah.....	3
1.3 Batasan Masalah	3
1.4 Tujuan	3
1.5 Manfaat	4
BAB II TINJAUAN PUSTAKA	5
2.1 Raspberry Pi 3.....	5
2.2 Raspbian.....	6
2.3 Bahasa Pemrograman Python	6
2.4 Database MySQL.....	8
2.5 Web Server.....	9
2.6 Internet	9
2.7 <i>Website</i>	10
2.8 Media Sosial.....	11
2.9 Google Drive.....	11
BAB III METODE PENELITIAN	13
3.1 Penentuan Spesifikasi Sistem	13
3.2 Perancangan dan Pembuatan Alat.....	13
3.2.1 Perancangan Perangkat Lunak.....	15
3.2.1.1 Konfigurasi Jaringan Internet dan <i>Web Server</i>	17
3.2.1.1.1 Konfigurasi Jaringan Internet.....	17
3.2.1.1.2 Konfigurasi <i>Web Server</i> Apache	18
3.2.1.2 Konfigurasi Database MySQL.....	19
3.2.1.3 <i>Script</i> PHP pada <i>Server</i>	23
3.2.1.3.1 <i>Script</i> PHP Penerima Data	23

3.2.1.3.2	<i>Script</i> PHP untuk Menampilkan Grafik	25
3.2.1.4	Bagian Pengolahan <i>Command user</i>	27
3.2.1.5	Integrasi Media Sosial.....	29
3.2.1.5.1	Integrasi Media Sosial WhatsApp.....	29
3.2.1.5.2	Integrasi Media Sosial Telegram.....	33
3.2.1.6	Integrasi Google Drive.....	35
3.3	Pengujian Alat.....	36
3.3.1	Pengujian <i>Server</i> Raspberry Pi 3.....	36
3.3.1.1	Pengujian RAM Raspberry Pi 3.....	36
3.3.1.2	Pengujian Suhu CPU Raspberry Pi 3.....	37
3.3.2	Pengujian Perangkat Lunak pada <i>Server</i>	37
3.3.2.1	Pengujian Pengiriman Data ke Database MySQL	37
3.3.2.2	Pengujian Kecepatan Pengiriman Data ke Database MySQL	37
3.3.2.3	Pengujian Pengolahan Database MySQL	37
3.3.2.4	Pengujian Menampilkan Data pada Grafik.....	37
3.3.2.5	Pengujian Pengiriman <i>Command</i> dari Media Sosial	38
3.3.2.6	Pengujian Pengiriman Data ke <i>User</i>	38
3.3.2.7	Pengujian Ukuran Data pada <i>Database</i>	38
3.3.2.8	Pengujian Penyimpanan <i>Database</i> pada Google Drive	38
3.3.2.9	Pengujian Command Switching.....	38
3.3.2.10	Pengujian Menggunakan Dua User.....	38
3.3.3	Pengujian Keseluruhan	38
BAB IV	HASIL DAN PEMBAHASAN	41
4.1	Pengujian <i>Server</i> Raspberry Pi 3	41
4.1.1	Pengujian RAM Raspberry Pi 3	41
4.1.2	Pengujian Suhu CPU Raspberry Pi 3	42
4.2	Pengujian Perangkat Lunak <i>Server</i>	42
4.2.1	Pengujian Data Pengiriman Sensor	42
4.2.2	Pengujian Kecepatan Pengiriman ke <i>Database</i>	44
4.2.3	Pengujian Pengolahan <i>Database</i> MySQL.....	45
4.2.4	Pengujian Menampilkan Data pada Grafik	45
4.2.5	Pengujian Pengiriman <i>Command</i> dari Media Sosial Telegram.....	46
4.2.6	Pengujian ukuran <i>file</i> *.csv dan lama waktu pengiriman	48

4.2.7	Penampil <i>file</i> *.csv.....	49
4.2.8	Pengujian Pengiriman Data Google Drive	49
4.2.9	Pengujian Dua <i>User</i>	51
4.2.10	Pengujian <i>Swithcing</i>	51
4.3	Pengujian Keseluruhan	52
BAB V KESIMPULAN DAN SARAN.....		53
5.1	Kesimpulan	53
5.2	Saran	53
DAFTAR PUSTAKA		55
LAMPIRAN		57

DAFTAR TABEL

Tabel 3. 1 Struktur Tabel <i>Monitoring Smart Grid</i>	23
Tabel 4. 1 Hasil pengujian RAM Raspberry Pi 3	42
Tabel 4. 2 Hasil pengujian suhu CPU Raspberry Pi 3.....	42
Tabel 4. 3 Data pengiriman modul sensor DHT node 1 ke database	43
Tabel 4. 4 Kecepatan pengiriman data ke database.....	44
Tabel 4. 5 Hasil Pengujian Pengiriman <i>Command</i>	47
Tabel 4. 6 Ukuran Memori <i>File *.csv</i> dan Lama Waktu Pengiriman	48
Tabel 4. 7 Pengiriman <i>database</i> pada google drive.....	50
Tabel 4. 8 <i>Delay</i> Pengiriman Data Dua <i>User</i>	51
Tabel 4. 9 Pengujian keseluruhan.....	52

DAFTAR GAMBAR

<i>Gambar 2. 1</i> Raspberry Pi 3 model B	6
<i>Gambar 2. 2</i> (a) Interpreter. (b) Kompiler.....	7
<i>Gambar 3. 1</i> Blok diagram perancangan sistem <i>switching</i> dan <i>monitoring smart grid</i>	14
<i>Gambar 3. 2</i> Blok diagram pengiriman data sensor hingga diterima <i>user</i>	15
<i>Gambar 3. 3</i> Blok diagram sistem <i>switching smart grid</i>	15
<i>Gambar 3. 4</i> Diagram alir perancangan <i>switching</i> dan <i>monitoring smart grid</i>	16
<i>Gambar 3. 5</i> Blok diagram pengiriman data melalui internet	17
<i>Gambar 3. 6</i> IP address Raspberry Pi 3	18
<i>Gambar 3. 7</i> Halaman web server Apache.....	19
<i>Gambar 3. 8</i> Konfigurasi PHPMyAdmin.....	20
<i>Gambar 3. 9</i> Konfigurasi PHPMyAdmin untuk membuat koneksi dengan <i>database</i>	20
<i>Gambar 3. 10</i> Membuat <i>password user</i> MySQL	21
<i>Gambar 3. 11</i> Repeat password MySQL.....	21
<i>Gambar 3. 12</i> Membuat <i>password user</i> phpmyadmin	21
<i>Gambar 3. 13</i> Repeat password phpmyadmin	21
<i>Gambar 3. 14</i> Password aplikasi MySQL untuk phpmyadmin	21
<i>Gambar 3. 15</i> Tampilan login PHPMyAdmin	22
<i>Gambar 3. 16</i> Diagram alir program <i>server</i> PHP menyimpan data ke <i>database</i> Mysql.....	24
<i>Gambar 3. 17</i> Listing program PHP menyimpan data ke <i>database</i>	25
<i>Gambar 3. 18</i> Diagram alir program <i>server</i> PHP penampil grafik	26
<i>Gambar 3. 19</i> Listing program PHP penampil grafik	27
<i>Gambar 3. 20</i> Diagram alir program <i>server</i> Python.....	28
<i>Gambar 3. 21</i> Tampilan yowsup berhasil mengirimkan kode verifikasi berupa pesan sms.....	30
<i>Gambar 3. 22</i> Tampilan pengiriman password akun WhatsApp	30
<i>Gambar 3. 23</i> Konfigurasi akun yowsup.....	31
<i>Gambar 3. 24</i> Permasalahan versi python-pip <i>six</i> 1.8.....	31
<i>Gambar 3. 25</i> Konfigurasi akun yowsup.....	32
<i>Gambar 3. 26</i> Tampilan <i>error</i> konfigurasi.....	33
<i>Gambar 3. 27</i> Tampilan detail akun yowsup WhatsApp	33

<i>Gambar 3. 28</i> Konfigurasi Bot API Telegram	34
<i>Gambar 3. 29</i> Identifikasi Bot API Telegram	35
<i>Gambar 3. 30</i> Otentikasi Google Drive.....	36
<i>Gambar 3. 31</i> Kode verifikasi Google Drive	36
<i>Gambar 3. 32</i> Penggunaan RAM Raspberry Pi 3	37
<i>Gambar 3. 33</i> Suhu CPU Raspberry Pi 3	37
<i>Gambar 4. 1</i> Tampilan database PHPMyAdmin.....	44
<i>Gambar 4. 2</i> Hasil Pembacaan <i>Database</i> Melalui Program Python	45
<i>Gambar 4. 3</i> Grafik <i>Monitoring Smart Grid</i>	46
<i>Gambar 4. 4</i> Jendela Program Python Shell.....	47
<i>Gambar 4. 5 Screenshot</i> Pengiriman Pesan dari Dua <i>User</i> Media Sosial Telegram.....	48
<i>Gambar 4. 6</i> Aplikasi Android Penampil File *.csv	49
<i>Gambar 4. 7</i> Pengiriman Data pada <i>Server</i>	50
<i>Gambar 4. 8</i> Data yang Tersimpan pada Google Drive	50
<i>Gambar 4. 9</i> ON/OFF Sistem Sesuai <i>Command User</i>	52

DAFTAR LAMPIRAN

Lampiran 1. Dokumentasi Alat	59
Lampiran 2. <i>Listing</i> Program.....	60
Lampiran 3. <i>Datasheet</i>	73

BAB I

PENDAHULUAN

1.1 Latar Belakang

Kemajuan teknologi akan berjalan sesuai dengan kemajuan ilmu pengetahuan. Setiap inovasi diciptakan untuk memberikan manfaat positif bagi kehidupan manusia. Teknologi juga memberikan banyak kemudahan, serta sebagai cara baru dalam melakukan aktivitas manusia. Salah satu pemanfaatan kemajuan teknologi adalah untuk berinovasi menciptakan sumber energi terbarukan dengan sistem yang canggih.

Energi merupakan salah satu kebutuhan dasar manusia. Saat ini kebutuhan energi di Indonesia terus mengalami peningkatan sejalan dengan tingkat kehidupannya, salah satunya adalah energi listrik. Energi listrik merupakan kebutuhan yang penting untuk membantu mempermudah aktivitas manusia. Perkembangan teknologi menunjukkan bahwa hampir seluruh peralatan rumah tangga, perkantoran, industri dan peralatan – peralatan lainnya menggunakan energi listrik. Penggunaan energi listrik terus mengalami peningkatan setiap tahunnya. Salah satu penyebabnya adalah penggunaan energi listrik yang tidak efisien khususnya pada sektor rumah tangga. Hal ini ditunjukkan oleh penjualan tenaga listrik PLN tahun 2015 naik sebesar 2,13% dibandingkan tahun 2014. Sedangkan produksi tenaga listrik PLN pada tahun 2015 naik sebesar 0,67% dibandingkan tahun 2014 (Dirjen Ketenagalistrikan, 2015).

Meninjau pola hidup masyarakat yang terus berkembang, peningkatan kebutuhan energi listrik menjadi sangat sulit dihindari. Di sisi lain peningkatan kebutuhan energi listrik tidak diimbangi dengan peningkatan ketersediaan energi listrik di Indonesia. Oleh karena itu, dibutuhkan sebuah solusi dengan memanfaatkan energi terbarukan.

Hingga saat ini sumber energi minyak bumi masih menjadi sumber energi utama didalam penggunaannya terutama dalam bidang kelistrikan, industri dan transportasi. Ditengah krisis energi saat ini timbul pemikiran untuk penganeekaragaman energi (diversifikasi energi) dengan mengembangkan sumber energi lain sebagai energi alternatif untuk penyediaan konsumsi energi domestik (Imam Kholiq, 2015). Indonesia memiliki potensi sumber daya energi yang beraneka ragam seperti energi air, angin, gelombang laut dan matahari, akan tetapi pemanfaatannya masih sedikit. Mengingat letak geografis Indonesia yang berada pada garis katulistiwa yang memungkinkan sinar matahari dapat

diterima hampir di seluruh wilayah Indonesia, maka pemanfaatan sumber daya energi terbarukan dapat dilakukan dengan menggunakan *smart grid*.

Smart grid adalah suatu jaringan listrik yang menggunakan teknologi digital dan teknologi maju lainnya untuk memantau dan mengelola transportasi listrik dari sumber pembangkitan listrik untuk memenuhi perubahan kebutuhan listrik dari pelanggan (BPPT, 2012). *Smart grid* disini dimanfaatkan sebagai sumber daya energi terbarukan untuk menyediakan energi listrik di Indonesia dengan memanfaatkan energi cahaya matahari yang diubah menjadi energi listrik. *Smart grid* merupakan suatu jaringan listrik yang telah terkomputerisasi dan terautomatisasi. Di dalam *Smart grid* memungkinkan komunikasi dua arah antara pengguna dan produsen. Selain itu transfer energi listrik yang terjadi tidak hanya dari produsen ke pengguna namun juga sebaliknya (Mochammad Choiril Iman, 2015).

Penggunaan *smart grid* sebagai sumber energi terbarukan dibutuhkan suatu sistem untuk *monitoring* serta proses *switching* pada sistem *smart grid* dari setiap *node*. Hal tersebut bertujuan untuk mengetahui apakah *smart grid* yang terpasang berjalan dengan baik atau tidak. Karena sistem keamanan menjadi kebutuhan yang mutlak untuk diterapkan dalam sistem instalasi listrik, maka dibutuhkan sistem yang dapat mengirimkan informasi mengenai kondisi sistem dengan fasilitas pengaksesan yang mudah. Proses *monitoring* dan proses *switching smart grid* dapat dilakukan secara *wireless* dengan menggunakan jaringan internet, sehingga proses *monitoring* dan proses *switching* dapat dilakukan dengan jangkauan jarak jauh melalui media sosial.

Proses *switching* dan *monitoring smart grid* jarak jauh berbasis internet dibutuhkan sebuah *server* yang berfungsi untuk menerima dan mengolah data pembacaan sensor - sensor yang terpasang pada setiap *node*, sehingga data dapat dikirim ke *user* melalui aplikasi media sosial. Salah satu aplikasi yang cukup banyak digemari dalam hal berbagi pesan adalah Whatsapp dan Telegram. Whatsapp dan Telegram adalah aplikasi pesan untuk *smartphone* yang merupakan aplikasi pesan lintas *platform* yang memungkinkan kita bertukar pesan tanpa biaya SMS, karena menggunakan paket data internet yang sama dengan Email, *browsing web* dan lain-lain (Fadli Sirait, 2015). Proses *switching* dan *monitoring smart grid* melalui media sosial yang menggunakan *server* Raspberry Pi 3 sebagai pemrosesnya.

Berdasarkan permasalahan tersebut peneliti akan merancang sebuah sistem *switching* dan *monitoring* parameter elektrik pada *smart grid* seperti tegangan dan arus berbasis Raspberry Pi 3 yang dapat diakses melalui media sosial. Skripsi ini merupakan pengembangan dan evaluasi terhadap skripsi yang ditulis oleh Machfud Firmansyah pada

tahun 2017 dengan judul “Sistem *Monitoring* Tegangan, Arus, dan Daya Pada *Smart Grid* Berbasis Raspberry Pi 3 Terintegrasi dengan Media Sosial Telegram dan *Website*”. Penulis menemukan beberapa kekurangan dalam proses *monitoring smart grid*, yaitu tidak dapat melakukan *pengontrolan* sistem saat ditemukan data yang tidak diharapkan pada saat proses pemantauan. Diharapkan skripsi yang disusun ini memberikan peningkatan dalam masalah yang ditemukan dalam perancangan sistem *switching* dan *monitoring smart grid* yang dapat dilakukan secara langsung pada waktu tertentu melalui media sosial.

1.2 Rumusan Masalah

Dari permasalahan yang mengacu pada latar belakang, maka rumusan masalah dapat ditekankan sebagai berikut:

1. Bagaimana perancangan sistem *switching* dan *monitoring* pada *smart grid* berbasis Raspberry Pi 3 yang terintegrasi dengan media sosial?
2. Bagaimana proses pengolahan informasi arus dan tegangan sehingga dapat dikirim dari pembacaan sensor ke media sosial?
3. Bagaimana proses *switching* sistem yang dapat dilakukan melalui pesan media sosial?

1.3 Batasan Masalah

Berdasarkan latar belakang di atas, maka dapat disusun rumusan masalah sebagai berikut:

1. *Server* menggunakan Raspberry Pi 3 mode B.
2. Sistem yang dirancang menggunakan tiga *node*.
3. Modul sensor masing-masing *node* tidak dibahas.
4. Sistem hanya menerima data dan mengolah data dari sensor.
5. Pengiriman data *monitoring* dalam batas waktu tertentu dikirim dalam format file *.csv.

1.4 Tujuan

Berdasarkan rumusan masalah tersebut maka tujuan dari penelitian ini adalah merancang sebuah sistem *switching* dan *monitoring* parameter elektrik seperti tegangan dan arus pada beberapa *node smart grid* berbasis Raspberry Pi 3 yang dapat diakses melalui media sosial.

1.5 Manfaat

Dari penelitian ini diharapkan dapat terciptanya sebuah alat *switching* dan *monitoring* tegangan dan arus pada *smart grid* yang dapat diakses secara langsung kapanpun dibutuhkan melalui pesan.

BAB II TINJAUAN PUSTAKA

2.1 Raspberry Pi 3

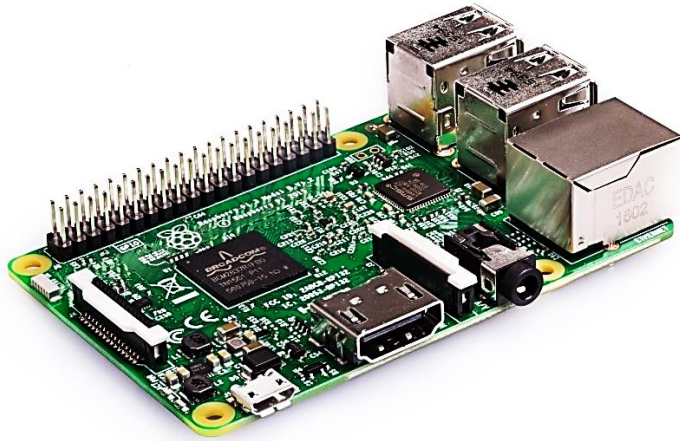
Raspberry Pi merupakan salah satu hasil pengembangan *single board computer*. Raspberry Pi adalah komputer kecil berukuran kartu kredit. Raspberry Pi dikembangkan di Inggris pada tahun 2011 oleh *Raspberry Pi Foundation* yang bertujuan untuk mempromosikan pengajaran dasar ilmu komputer. Komputer mini ini mampu bekerja layaknya PC Str dengan kemampuan untuk menjalankan OS Linux dan aplikasinya, seperti Multimedia (*Audio, Video, Picture*), *Programming* (QT, Python, C++), *database server*, dll. Raspberry Pi juga dapat menampilkan gambar ke TV HDTV dengan koneksi HDMI ataupun TV Strd dengan koneksi TV *Out* (Therzian Richard Perkasa, Helmy Widyantara, pauladie Susanto, 2014).

Raspberry Pi memiliki sistem *Broadcom BCM2835 chip* (SoC), yang mencakup ARM1176JZF-S 700 MHz *processor*, VideoCore IV GPU, dan awalnya dikirim dengan 256 megabyte RAM, kemudian *upgrade* ke 512MB. Termasuk *built-in hard disk* atau *solid-state drive*, tetapi menggunakan kartu SD untuk *booting* dan penyimpanan jangka panjang (Malik Abdillah Ibnu Hakim, Yeffry Handoko Putra, 2013).

Adapun spesifikasi Raspberry Pi model-B sebagai berikut:

- a. Catu daya : 5 VDC, 700 mA (via *micro USB*)
- b. Berbasis mikrokontroler/mikroprosesor : ARM1176JZF-S core, 700 MHz
- c. *Port* antarmuka : UART TTL, SPI, I2C, USB, *Composite RCA*, 3.5 mm *jack*, 10/100 *Ethernet* (RJ45), LCD *Panels* via DSI, CSI(*Camera Serial Interface*), HDMI
- d. *Bootloader* : melalui OS berbasis LINUX
- e. Fitur : *Memory* 512 MB, 4 USB PORT, *Graphics Broadcom VideoCore IV* , SD *Card Slot*
- f. Dimensi : 85.60mm(L) x 56mm(W) x 21mm(H)

(Therzian Richard Perkasa, Helmy Widyantara, pauladie Susanto, 2014).



Gambar 2. 1 Raspberry Pi 3 model B
Sumber: Raspberry Pi (2017)

2.2 Raspbian

Untuk menjalankan fungsi Raspberry Pi secara keseluruhan, dibutuhkan sistem operasi yang mendukung fungsi dan performa Raspi, salah satunya adalah Raspbian. Raspbian merupakan sistem operasi yang berbasis Debian khusus untuk Raspberry Pi. Untuk para pemula, dianjurkan untuk menggunakan Raspbian sebagai sistem operasi awal, karena proses instalasinya yang mudah dan menyediakan lebih dari 35.000 paket *software* untuk mengoptimalkan kinerja Raspberry Pi. Raspbian kini masih dalam perkembangan aktif, dan lebih ditekankan untuk peningkatan stabilitas dan kinerja dengan paket Debian sebanyak mungkin (Edi Rakhman, 2014).

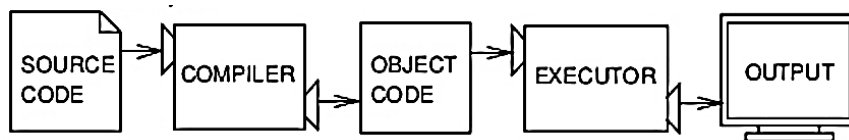
2.3 Bahasa Pemrograman Python

Python merupakan bahasa pemrograman yang *freeware* atau perangkat bebas dalam arti sebenarnya, tidak ada batasan dalam penyalinannya atau mendistribusikannya. Lengkap dengan *source code*-nya, *debugger* dan *profiler*, antarmuka yang terkandung di dalamnya untuk pelayanan antarmuka, fungsi sistem, GUI (antarmuka pengguna grafis), dan basis datanya. Python menjadi bahasa resmi yang terintegrasi dalam Raspberry Pi. Kata “Pi” pada Raspberry Pi merupakan slang yang merujuk pada “Python”. Oleh karenanya, tepat dikatakan bahwa Python adalah bahasa natural Raspberry Pi (Edi Rakhman, 2014).

Terdapat dua jenis aplikasi untuk memproses bahasa tingkat tinggi ke bahasa tingkat rendah, yaitu *compiler* dan *interpreter*. Sebuah *interpreter* membaca sebuah program yang ditulis dengan bahasa tingkat tinggi dan langsung menjalankannya per baris, memakan waktu sedikit. Sedangkan sebuah *kompiler* menerjemahkan keseluruhan kode program sebelum menjalankan program tersebut (Allen Downey, 2012).



(a)



(b)

Gambar 2. 2 (a) Interpreter. (b) Kompiler

Sumber: Allen Downey (2012)

Beberapa fitur yang dimiliki Python adalah:

1. Memiliki kepustakaan yang luas; dalam distribusi Python telah disediakan modul–modul siap pakai untuk berbagai keperluan.
2. Memiliki tata bahasa yang jernih dan mudah dipelajari.
3. Memiliki aturan *layout* kode sumber yang memudahkan pengecekan, pembacaan kembali, dan penulisan ulang kode sumber berorientasi objek.
4. Memiliki sistem pengelolaan memori otomatis (gerbang *collection*, seperti Java)
5. Modular, mudah dikembangkan dengan menciptakan modul – modul baru. Modul – modul tersebut dapat dibangun dengan bahasa Python maupun C/C++.
6. Memiliki fasilitas pengumpulan sampah otomatis. Seperti halnya pada bahasa pemrograman Java, Python memiliki fasilitas pengaturan penggunaan ingatan komputer sehingga para pemrograman tidak perlu melakukan pengaturan ingatan komputer secara langsung.

Python memiliki kelebihan sebagai berikut:

- a. Tidak ada tahapan kompilasi dan penyambungan (*link*) sehingga kecepatan perubahan pada masa pembuatan sistem aplikasi meningkat.
- b. Tidak ada deklarasi tipe sehingga program menjadi lebih sederhana, singkat dan fleksibel.
- c. Manajemen memori otomatis, yaitu kumpulan sampah memori sehingga dapat menghindari pencatatan kode.
- d. Tipe data dan operasi tingkat tinggi, yaitu kecepatan pembuatan sistem aplikasi menggunakan tipe objek yang telah ada.
- e. Pemrograman berorientasi objek.

- f. Pelekatan dan perluasan dalam C.
- g. Terdapat kelas, modul, eksepsi, sehingga terdapat dukungan pemrograman skala besar secara modular.
- h. Pemuatan dinamis modul C sehingga ekstensi menjadi sederhana dan berkas biner yang kecil.
- i. Pemuatan kembali secara dinamis modul Python, seperti memodifikasi aplikasi tanpa menghentikannya.
- j. Model objek universal kelas satu.
- k. Konstruksi saat aplikasi berjalan.
- l. Interaktif, dinamis dan alamiah.
- m. Akses hingga informasi interpreter.
- n. Portabilitas secara luas seperti pemrograman antar *platform* tanpa *ports*.
- o. Kompilasi untuk *portable code byte* sehingga kecepatan eksekusi bertambah dan melindungi kode sumber.
- p. Antarmuka terpasang untuk pelayanan keluar, seperti perangkat bantu sistem, GUI, *persistence*, *database* dan lain – lain (Edi Rakhman, 2014).

2.4 Database MySQL

Database adalah kumpulan informasi yang disimpan di dalam komputer secara sistematis sehingga dapat diperiksa menggunakan suatu program komputer untuk memperoleh informasi dari basis data tersebut (Alhadi Saputra, 2012). MySQL adalah suatu perangkat lunak *database* relasi (*Relational Database Management System* atau DBMS), seperti halnya ORACLE, POSTGRESQL, MSSQL, dan sebagainya. SQL merupakan singkatan dari *Structure Query Language*, didefinisikan sebagai suatu sintaks perintah - perintah tertentu atau bahasa program yang digunakan untuk mengelola suatu *database*. Jadi MySQL adalah *software*nya dan SQL adalah bahasa perintahnya (Anisya, 2013).

MySQL adalah aplikasi *database* yang berjalan sebagai *service*. Aplikasi *service* berjalan tanpa menampilkan antarmuka pada *desktop* atau *taskbar*. MySQL dijalankan pada *mode Text* atau *Command prompt* atau dengan menggunakan PHPMysqlAdmin. PHPMysqlAdmin merupakan aplikasi *web* yang dapat digunakan untuk manajemen dan administrasi *server* dan *database* serta obyek-obyek yang terdapat di dalamnya. Proses menggunakan MySQL pada dasarnya adalah mengelola data dan informasi agar data dan informasi tersimpan dengan tertata, proses-proses yang sering terjadi biasanya adalah membuat *database*, membuat sebuah tabel, memodifikasi struktur sebuah tabel, mengisi

data dalam sebuah tabel, memodifikasi (merubah atau mengedit) data dalam sebuah tabel dan mencari data dalam sebuah tabel (Alhadi Saputra, 2012).

HP (*Hypertext Preprocessor*) adalah bahasa skrip yang dapat ditanamkan atau disisipkan ke dalam HTML. PHP banyak dipakai untuk memrogram situs *web* dinamis. PHP dapat digunakan untuk membangun sebuah CMS. Pada awalnya PHP merupakan singkatan dari *Personal Home Page* (Situs personal). Berikut adalah contoh koneksi PHP dengan *Database*:

1. Jika konek ke *database* berhasil, perlu memilih *database*.
2. Perintah untuk memilih *database*:

mysql_select_db(data_base,pengenal_hubungan)

dimana: **data_base** = nama *database*

pengenal_hubungan = nama pengenal yang digunakan dalam koneksi

(Anisa, 2013)

2.5 Web Server

Web server adalah suatu *server* yang menyimpan halaman-halaman *web* dari suatu instansi atau perusahaan tertentu yang dapat diakses dengan *web browser*. *Web server* diciptakan oleh suatu *software* aplikasi, salah satunya adalah Apache. Apache adalah *web server* yang dapat dijalankan pada banyak sistem operasi (Unix, BSD, Linux, Microsoft Windows dan Novell Netware serta *platform* lainnya) yang berguna untuk melayani dan memfungsikan situs *web*. Apache berkembang dengan cepat dan banyak digunakan karena mendukung bahasa *scripting*, seperti PERL dan PHP, dan juga merupakan produk *open source*. Apache bekerja dengan menggunakan *Hyper Text Transfer Protocol*. Sehingga Apache sering disebut dengan *httpd*, yang berarti daemon *http* (Iwan Binanto, 2005).

2.6 Internet

Internet merupakan jaringan komputer yang saling terhubung menggunakan standar sistem global *Transmission Control Protocol/Internet Protocol Suite* (TCP/IP) sebagai protokol pertukaran paket (*packet switching communication protocol*) untuk melayani miliaran pengguna di seluruh dunia. Rangkaian internet yang terbesar dinamakan Internet., sehingga pengguna bisa mendapatkan informasi dari seluruh dunia serta bisa digunakan untuk berkirim surat (e-mail), transfer data, serta chatting. Pada umumnya fasilitas ini sering digunakan untuk berkomunikasi di internet *World Wide Web* (WWW) yang memberikan kemudahan untuk mengambil, memformat, dan menampilkan informasi termasuk teks, audio, grafik, dan video berdasarkan standar Internet Protocol (IP) (Juli Arianes, 2016).

Jaringan yang membentuk internet bekerja berdasarkan suatu set protokol standar yang digunakan untuk menghubungkan jaringan komputer dan mengamati lalu lintas dalam jaringan. Protokol ini mengatur format data yang diijinkan, penanganan kesalahan (*error handling*), lalu lintas pesan, dan standar komunikasi lainnya. Protokol standar pada internet dikenal sebagai TCP/IP (*Transmission Control Protocol/Internet Protocol*). Protokol ini memiliki kemampuan untuk bekerja diatas segala jenis komputer, tanpa terpengaruh oleh perbedaan perangkat keras maupun sistem operasi yang digunakan. Sebuah sistem komputer yang terhubung secara langsung ke jaringan memiliki nama domain dan alamat IP (*Internet Protocol*) dalam bentuk numerik dengan format tertentu sebagai pengenalan. Internet juga memiliki *gateway* ke jaringan dan layanan yang berbasis protokol lainnya (Graifhan Ramadhani, 2003).

2.7 Website

Website adalah kumpulan dan halaman-halaman situs, yang biasanya terangkum dalam sebuah domain dan sub domain, yang tempatnya berada di dalam *World Wide Web* (WWW) di internet. WWW atau yang sering disebut sebagai "*web*" merupakan aplikasi internet yang paling populer. Demikian populernya hingga banyak orang yang keliru mengidentikkan *web* dengan internet. Secara teknis, *web* adalah sebuah sistem dimana informasi dalam bentuk teks, gambar, suara, dan lain-lain yang tersimpan dalam sebuah internet *webserver* dipresentasikan dalam bentuk *hypertext*. Informasi di *web* dalam bentuk teks umumnya ditulis dalam format HTML (*Hypertext Markup Language*). Informasi lainnya disajikan dalam bentuk grafis (dalam format GIF, JPG, PNG), suara (dalam format AU, WAV), dan objek multimedia lainnya (seperti MIDI, *Shockwave*, *Quicktime Movie*, 3D *World*). *Web* dapat diakses oleh perangkat lunak *web client* yang secara populer disebut sebagai *browser*. *Browser* membaca halaman-halaman *web* yang tersimpan dalam *webserver* melalui protokol yang disebut HTTP (*Hypertext Transfer Protocol*). Sebagai dokumen *hypertext*, dokumen-dokumen di *web* dapat memiliki *link* (sambungan) dengan dokumen lain, baik yang tersimpan dalam *webserver* yang sama maupun di *webserver* lainnya. *Link* memudahkan para pengakses *web* berpindah dari satu halaman ke halaman lainnya, dan "berkelana" dari satu *server* ke *server* lain. Untuk memudahkan penelusuran halaman *web*, terutama untuk menemukan halaman yang memuat topik-topik yang spesifik, maka para pengakses *web* dapat menggunakan suatu *search engine* (mesin pencari). Penelusuran berdasarkan *search engine* dilakukan berdasarkan kata kunci (*keyword*) yang kemudian akan dicocokkan oleh *search engine* dengan *database* (basis data) miliknya (Graifhan Ramadhani, 2003).

2.8 Media Sosial

Media sosial merupakan aplikasi berbagi pesan tanpa menggunakan biaya pulsa, penggunaannya hanya membutuhkan koneksi internet. Saat ini media sosial terus mengalami pengembangan dalam fitur-fiturnya. Tidak hanya digunakan untuk berbagi pesan tertulis saja, media sosial juga telah mampu melakukan panggilan, berbagi gambar, *video*, pesan suara, bahkan panggilan *video*.

Salah satu aplikasi yang cukup banyak digemari dalam hal berbagi pesan adalah *Whatsapp Messenger*. Dengan aplikasi pesan *Whatsapp* memungkinkan pengguna untuk mengirim pesan rahasia yang diekripsi *end-to-end* sebagai keamanan tambahan. *Whatsapp* adalah aplikasi pesan untuk *smartphone* yang merupakan aplikasi pesan lintas *platform* yang memungkinkan kita bertukar pesan tanpa biaya SMS, karena *Whatsapp* menggunakan paket data internet yang sama dengan email, *browsing web* dan lain-lain (Fadli Sirait, 2015)

Selain itu, aplikasi media sosial yang tidak kalah populer adalah *Telegram*. *Telegram* adalah sebuah aplikasi berkirim pesan dengan fokus kecepatan dan keamanan, aplikasi ini super cepat, sederhana, dan gratis. *Telegram* dapat digunakan pada banyak perangkat pada waktu bersamaan. Semua pesan tersinkronisasi secara otomatis ke semua perangkat baik *handphone*, *tablet*, maupun komputer yang terkoneksi ke *Telegram* (*Telegram*, 2016).

2.9 Google Drive

Google Drive adalah layanan penyimpanan *online* milik *Google*. Layanan ini merupakan ekstensi dari *Google Docs* dan akan mengganti URL *docs.google.com* dengan *drive.google.com* setelah diaktifkan. *Google Drive* memberikan layanan penyimpanan gratis sebesar 15 GB dan dapat ditambahkan dengan pembayaran tertentu. Dengan fitur unggulan yang sama seperti *Dropbox*, yaitu sinkronisasi data melalui berkas khusus di dalam desktop atau lebih dikenal dengan *Desktop Sync Clients*. *Google Drive* memberikan kapasitas gratis sebesar 15 GB dan tentunya fitur-fitur yang terintegrasi dengan layanan *Google* lainnya seperti: *Gmail*, *G+*, dan *Google Search* (*Google*, 2016).

BAB III METODE PENELITIAN

Sistem *switching* dan *monitoring smart grid* jarak jauh berbasis internet dibutuhkan sebuah *server* yang berfungsi untuk menerima dan mengolah data pembacaan sensor - sensor yang terpasang pada setiap *node*, sehingga data dapat dikirim ke *user* melalui aplikasi media sosial. Proses *switching* dan *monitoring smart grid* dilakukan melalui pengiriman pesan media sosial, kemudian sebagai pemrosesnya digunakan minikomputer Raspberry Pi 3. Penyimpanan data *monitoring* disimpan secara rutin pada sebuah akun google drive, sehingga memungkinkan untuk menampilkan data dari jarak jauh.

Metode penelitian yang dilakukan untuk perancangan sistem tersebut bersifat aplikatif. Langkah-langkah yang perlu dilakukan adalah penentuan spesifikasi sistem, perancangan dan pembuatan alat, pengujian alat dan pengambilan kesimpulan.

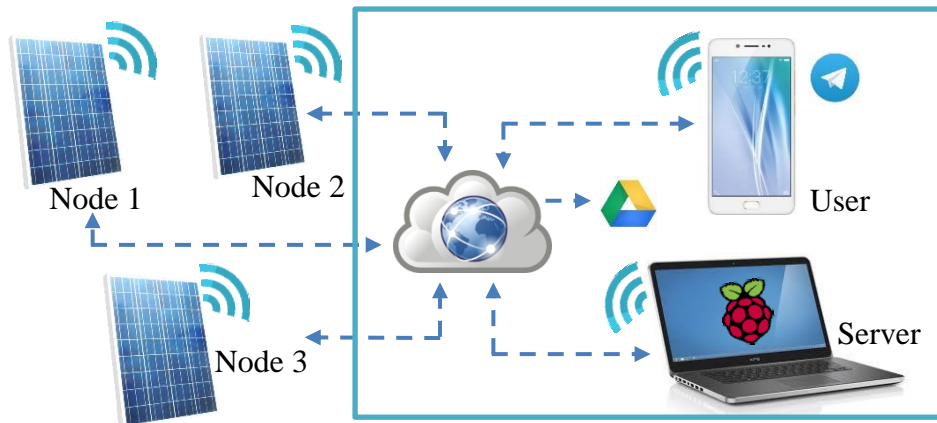
3.1 Penentuan Spesifikasi Sistem

Berdasarkan perancangan sistem *monitoring* dan *switching smart grid* tersebut, digunakan spesifikasi sistem sebagai berikut:

1. Raspberry Pi 3 digunakan sebagai *server* yang menerima dan mengolah data hasil pembacaan sensor.
2. Bahasa pemrograman yang digunakan yaitu Python dengan sistem operasi Raspbian.
3. Data hasil pembacaan sensor disimpan dalam bentuk tabel pada *database* MySQL.
4. Sistem *switching* dan *monitoring* menggunakan media sosial.
5. Proses *switching* dan pengiriman data *monitoring* dilakukan dengan mengirimkan *command* tertentu dari media sosial ke *server*.
6. Data *monitoring* dikirimkan dalam batas waktu tertentu (*real time*) yang dikirimkan dalam format *.csv.

3.2 Perancangan dan Pembuatan Alat

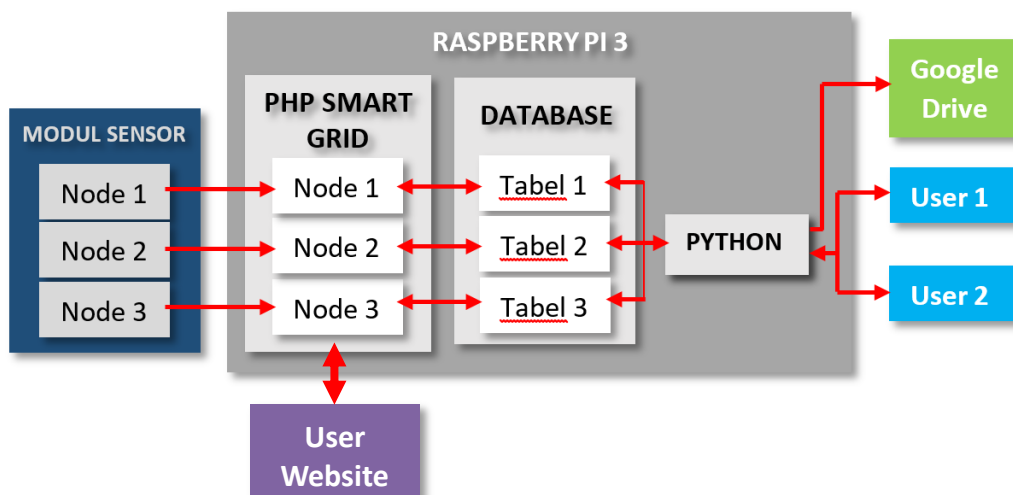
Perancangan alat pada sistem *monitoring* dan *switching smart grid* meliputi perancangan perangkat lunak untuk menjalankan sistem. Berikut adalah blok diagram sistem *switching* dan *monitoring smart grid*.



Gambar 3. 1 Blok diagram perancangan sistem *switching* dan *monitoring smart grid*

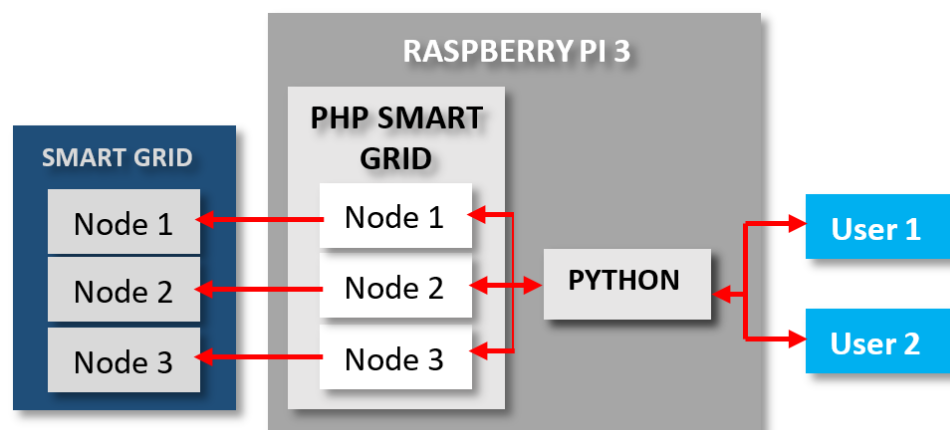
Perancangan dan penelitian yang dilakukan adalah proses pengolahan informasi data tegangan dan arus sehingga dapat ditampilkan pada media sosial tanpa membahas proses pada masing-masing *node*. Sistem tersebut melibatkan sebuah *server* yaitu Raspberry Pi 3 dan internet sebagai pusat penerimaan dan pengiriman data sehingga dapat dilakukan dalam jarak jauh. Sehingga dalam blok diagram perancangan sistem *switching* dan *monitoring smart grid* tersebut hanya ditekankan pada sistem selain perancangan pada masing-masing *node* dengan berbagai sensor dan perangkat yang digunakan sehingga dapat mengirimkan informasi tegangan dan arus secara *wireless*.

Proses pengiriman data masing-masing blok dilakukan secara *wireless*, dalam perancangan sistem *switching* dan *monitoring smart grid* digunakan tiga *node*. Masing-masing *node* akan mengirimkan data pembacaan nilai tegangan dan arus pada *database* yang telah dibuat. *Monitoring smart grid* dilakukan melalui media sosial melalui program Python. Program Python akan mengirimkan *file* yang dapat dibuka dengan aplikasi android seperti Excel. Program Python juga akan menyimpan database pada Google Drive sehingga dapat diakses kapanpun dan dimanapun. Berikut adalah blok diagram pengiriman data dari *node* hingga tampil pada *user*.



Gambar 3. 2 Blok diagram pengiriman data sensor hingga diterima user

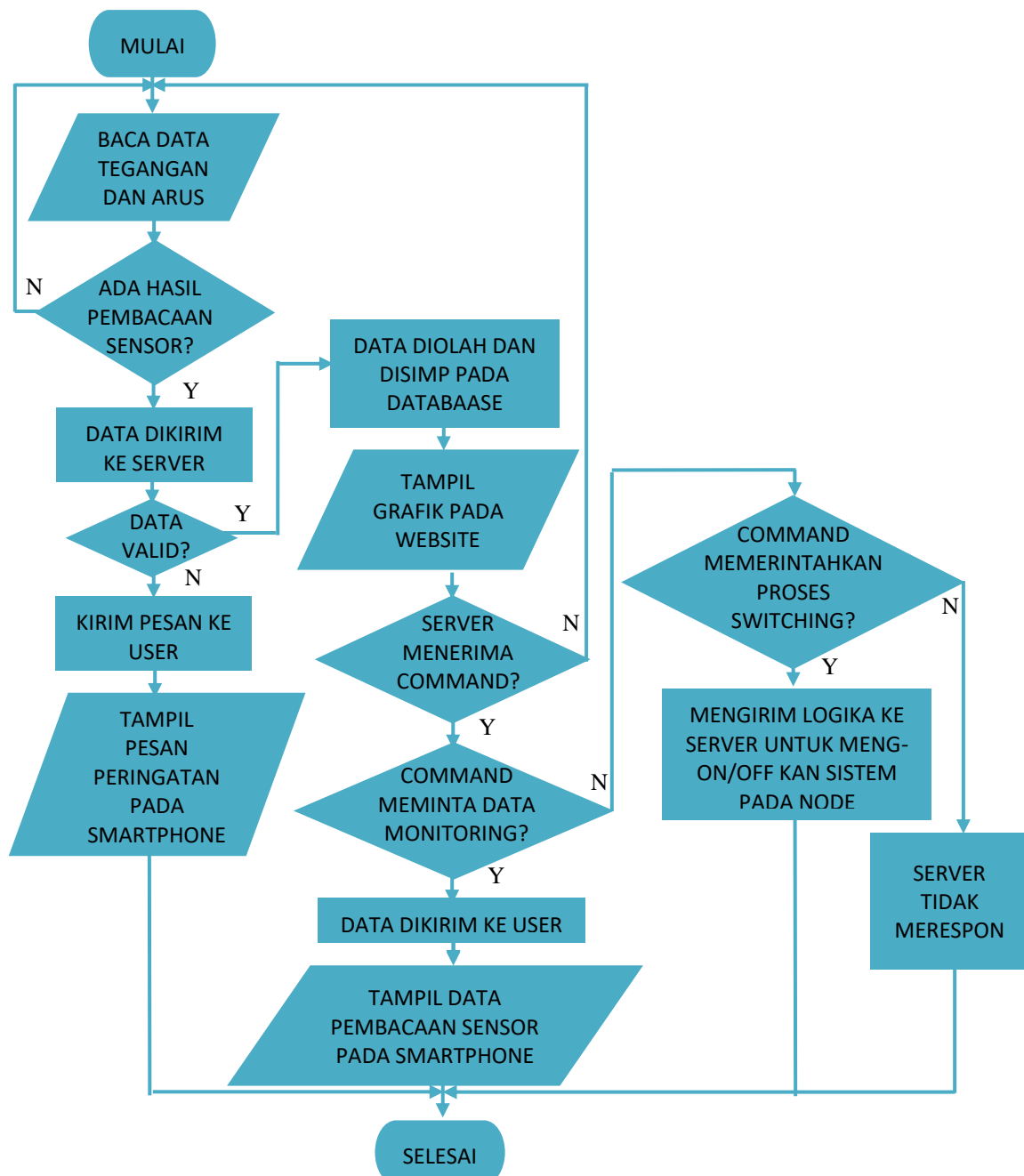
Proses *switching* pada *smart grid* bertujuan untuk menyambungkan atau memutuskan aliran listrik dari sumber, hal tersebut dibutuhkan untuk penanganan cepat saat terjadi kondisi darurat. Proses *switching* dilakukan dengan mengirimkan pesan pada media sosial berupa perintah *ON* atau *OFF* untuk *node* yang dituju. Python memproses pesan yang diterima kemudian mengirimkan sebuah perintah pada *node* yang dituju. Berikut adalah blok diagram proses *switching*.



Gambar 3. 3 Blok diagram sistem *switching smart grid*

3.2.1 Perancangan Perangkat Lunak

Perancangan perangkat lunak pada sistem *monitoring* dan *switching* memerlukan beberapa tahap untuk menjalankan kerja alat secara keseluruhan. Berikut gambar diagram alir perangkat lunak sistem *monitoring* dan *switching smart grid*.



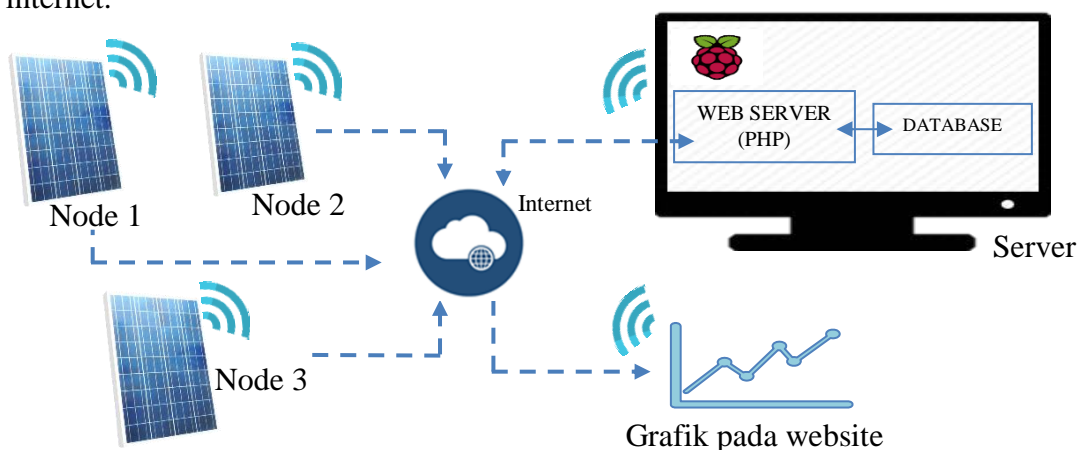
Gambar 3. 4 Diagram alir perancangan *switching* dan *monitoring smart grid*

Gambar 3.4 menunjukkan diagram alir sistem *switching* dan *monitoring smart grid* secara keseluruhan yang akan mengirimkan informasi nilai tegangan dan arus dari modul-modul sensor *smart grid* kepada *server*. Data pembacaan yang *valid* akan diolah oleh *server* dengan menggunakan Raspberry Pi dengan menyimpannya ke dalam *database MySQL* dan menampilkannya dalam bentuk grafik pada *website*. *Server* juga dapat menerima *command* dari *user* dengan format tertentu, apabila *command* yang dikirimkan salah, maka *server* tidak akan merespon. Sedangkan ketika *command* yang dikirim meminta data *monitoring*, maka informasi nilai pembacaan modul sensor pada masing-masing *node* akan dikirim ke *user*

dalam bentuk *file *.csv* melalui media sosial sesuai dengan rentang waktu yang diminta oleh *user*. Sedangkan ketika *command* yang dikirim meminta proses *switching*, maka *server* akan mengirimkan perintah kepada *node* sensor yang dituju untuk melakukan perintah *ON* atau *OFF*.

3.2.1.1 Konfigurasi Jaringan Internet dan Web Server

Proses pengiriman dari modul sensor pada *node smart grid* ke *server* Raspberry Pi membutuhkan jaringan internet. Berikut adalah Blok diagram pengiriman data melalui internet.



Gambar 3. 5 Blok diagram pengiriman data melalui internet

Ketika terdapat nilai tegangan dan arus pada pembacaan modul sensor, maka nilai-nilai tersebut akan dikirim secara *wireless* melalui internet dengan menggunakan WiFi dengan alamat tujuan adalah alamat IP Raspberry Pi. Nilai tegangan dan arus akan dibaca oleh program PHP dan disimpan ke database MySQL. Data yang disimpan pada *database* juga akan ditampilkan dalam bentuk grafik garis yang dapat diakses melalui alamat IP Raspberry Pi dengan diikuti nama file PHP. Grafik menampilkan data tiga node sekaligus dalam satu laman. Konfigurasi yang dibutuhkan untuk sistem ini antara lain konfigurasi jaringan internet, konfigurasi *web server*, konfigurasi *database*, dan konfigurasi file PHP.

3.2.1.1.1 Konfigurasi Jaringan Internet

Raspberry Pi 3 digunakan sebagai *server* utama dalam perancangan *monitoring smart grid*. Proses pengiriman data secara *wireless* dibutuhkan alamat IP dari *server* Raspberry PI agar data yang dikirimkan dapat diterima oleh *server*. Untuk mengetahui alamat IP dari Raspberry Pi 3 yaitu dengan mengetikkan *command* pada terminal sebagai berikut:

```
pi@raspberrypi:~ $ sudo ifconfig
```



```

eth0      Link encap:Ethernet  HWaddr b8:27:eb:85:12:a1
          inet6 addr: fe80::c3e:771e:8b26:5f80/64 Scope:Link
          UP BROADCAST MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
          RX packets:212 errors:0 dropped:0 overruns:0 frame:0
          TX packets:212 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1
          RX bytes:17508 (17.0 KiB)  TX bytes:17508 (17.0 KiB)

wlan0     Link encap:Ethernet  HWaddr b8:27:eb:d0:47:f4
          inet addr:192.168.0.104 Bcast:192.168.0.255  Mask:255.255.255.0
          inet6 addr: fe80::c113:e080:5e61:4455/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:22 errors:0 dropped:0 overruns:0 frame:0
          TX packets:43 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:2530 (2.4 KiB)  TX bytes:7303 (7.1 KiB)

```

Gambar 3. 6 IP address Raspberry Pi 3

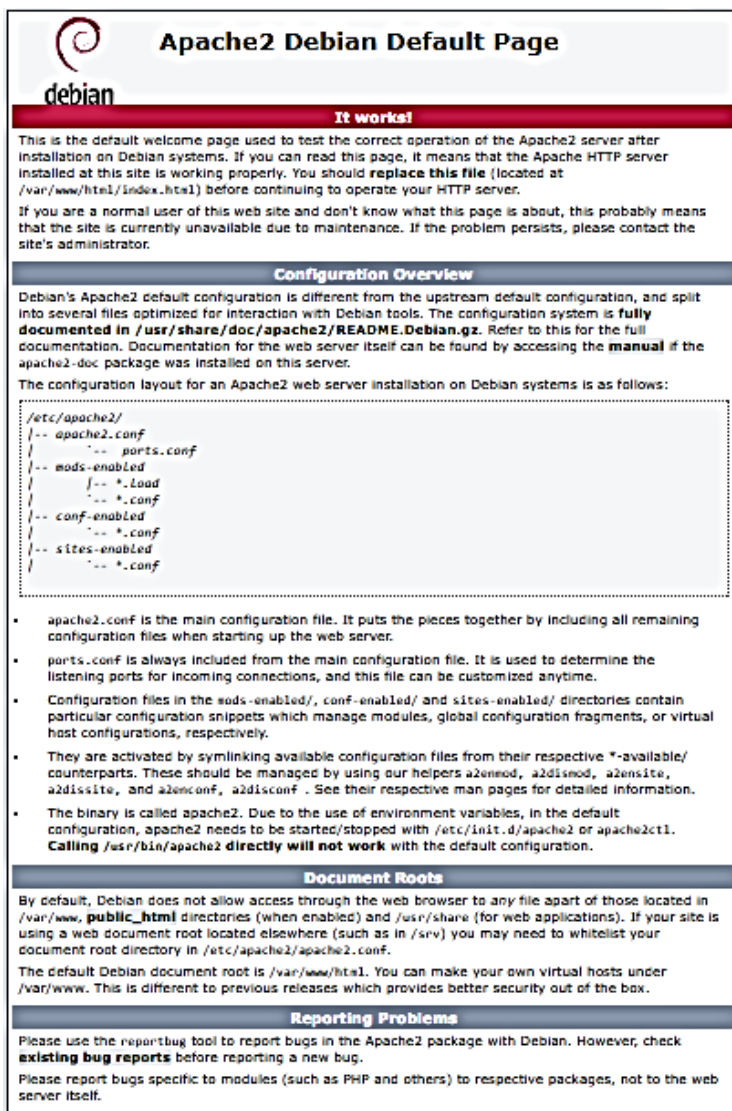
Alamat IP Raspberry Pi digunakan sebagai alamat untuk mengirimkan data dari modul sensor yang terpasang pada masing-masing *node*, serta untuk membuka halaman *web* yang akan menampilkan informasi nilai tegangan dan arus pada *smart grid* dalam bentuk grafik dengan mengetikkan alamat IP Raspberry Pi dan nama *file* PHP yang dibuat pada kolom mesin telusur. Grafik dapat ditampilkan pada berbagai macam *web browser* seperti Chrome, Mozilla Firefox, Internet Explorer dan lain sebagainya. IP *address* yang digunakan dalam perancangan yaitu IP *address* dinamis, sehingga ketika pertama kali Raspberry Pi dinyalakan maka alamat IP tersebut akan berubah-ubah.

3.2.1.1.2 Konfigurasi Web Server Apache

Web server Apache berfungsi untuk menjalankan program PHP agar dapat menerima data dari modul-modul sensor melalui internet dan menyimpannya ke dalam *database* MySQL. Untuk melakukan konfigurasi *web server* Apache yaitu dengan mengetikkan *command* sebagai berikut pada terminal:

```
pi@raspberrypi:~ $ sudo apt-get install apache2 -y
```

Setelah proses install *web server* Apache selesai, maka dengan menggunakan IP *address* Raspberry Pi akan tampil *web page* dengan *content default* dari *web server* Apache seperti gambar 3.7 yang menunjukkan bahwa *web server* telah berhasil dijalankan.



Apache2 Debian Default Page

It works!

This is the default welcome page used to test the correct operation of the Apache2 server after installation on Debian systems. If you can read this page, it means that the Apache HTTP server installed at this site is working properly. You should **replace this file** (located at `/var/www/html/index.html`) before continuing to operate your HTTP server.

If you are a normal user of this web site and don't know what this page is about, this probably means that the site is currently unavailable due to maintenance. If the problem persists, please contact the site's administrator.

Configuration Overview

Debian's Apache2 default configuration is different from the upstream default configuration, and split into several files optimized for interaction with Debian tools. The configuration system is **fully documented in `/usr/share/doc/apache2/README.Debian.gz`**. Refer to this for the full documentation. Documentation for the web server itself can be found by accessing the **manual** if the `apache2-doc` package was installed on this server.

The configuration layout for an Apache2 web server installation on Debian systems is as follows:

```

/etc/apache2/
|-- apache2.conf
|   |-- ports.conf
|-- mods-enabled
|   |-- *.load
|   |-- *.conf
|-- conf-enabled
|   |-- *.conf
|-- sites-enabled
|   |-- *.conf

```

- `apache2.conf` is the main configuration file. It puts the pieces together by including all remaining configuration files when starting up the web server.
- `ports.conf` is always included from the main configuration file. It is used to determine the listening ports for incoming connections, and this file can be customized anytime.
- Configuration files in the `mods-enabled/`, `conf-enabled/` and `sites-enabled/` directories contain particular configuration snippets which manage modules, global configuration fragments, or virtual host configurations, respectively.
- They are activated by symlinking available configuration files from their respective `*-available/` counterparts. These should be managed by using our helpers `a2enmod`, `a2dismod`, `a2ensite`, `a2dissite`, and `a2enconf`, `a2disconf`. See their respective man pages for detailed information.
- The binary is called `apache2`. Due to the use of environment variables, in the default configuration, `apache2` needs to be started/stopped with `/etc/init.d/apache2` or `apache2ctl`. Calling `/usr/bin/apache2` directly **will not work** with the default configuration.

Document Roots

By default, Debian does not allow access through the web browser to any file apart of those located in `/var/www`, **public_html** directories (when enabled) and `/usr/share` (for web applications). If your site is using a web document root located elsewhere (such as in `/srv`) you may need to whitelist your document root directory in `/etc/apache2/apache2.conf`.

The default Debian document root is `/var/www/html`. You can make your own virtual hosts under `/var/www`. This is different to previous releases which provides better security out of the box.

Reporting Problems

Please use the `reportbug` tool to report bugs in the Apache2 package with Debian. However, check **existing bug reports** before reporting a new bug.

Please report bugs specific to modules (such as PHP and others) to respective packages, not to the web server itself.

Gambar 3. 7 Halaman web server Apache

Setelah proses instalasi *web server* selesai, dilanjutkan dengan meng-*install* PHP supaya dapat membuat *listing* program dalam *file* PHP yang akan digunakan untuk menerima data sensor dari masing- masing modul sensor serta menampilkan grafik. *Command* untuk *install* PHP dapat diketik seperti berikut:

```
pi@raspberrypi:~$ sudo apt-get install php5 libapache2-mod-php5 -y
```

3.2.1.2 Konfigurasi Database MySQL

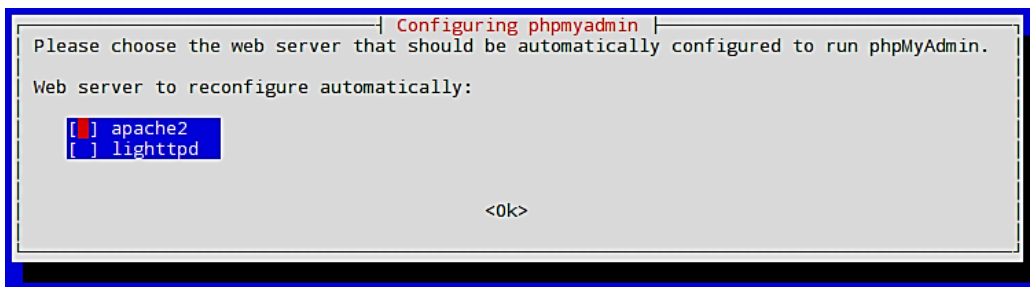
Sistem *monitoring smart grid* dibutuhkan *database* yang berfungsi untuk menyimpan data-data yang berisi nilai tegangan dan arus dari setiap *node*. Untuk melakukan konfigurasi *database* MySQL yaitu dengan mengetikkan *command* sebagai berikut pada terminal:

```
pi@raspberrypi:~$ sudo apt-get install mysql-server php5-mysql
```

Database MySQL dapat dijalankan secara *online* dengan phpmyadmin. Penggunaan phpmyadmin untuk akses *database* sangat memudahkan *user* untuk mengelola *database*, sebab tidak perlu mengetikkan *query* untuk melakukan suatu perintah. Proses *install* phpmyadmin dilakukan dengan mengetikkan *command* sebagai berikut:

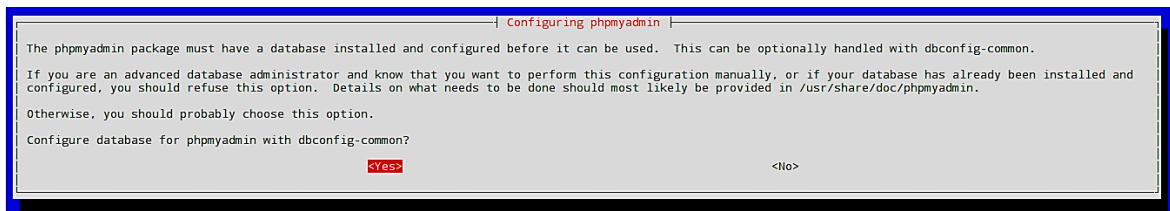
```
pi@raspberrypi:~ $ sudo apt-get install phpmyadmin
```

Selama proses instalasi, akan muncul permintaan untuk memilih *web server* yang akan digunakan untuk mengakses *database* MySQL secara *online*. Maka pilih Apache2 sesuai dengan *web server* yang telah di-*install*.



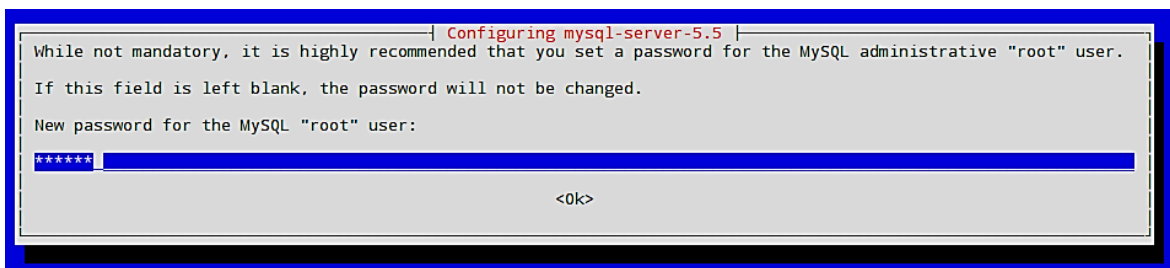
Gambar 3. 8 Konfigurasi PHPMyAdmin

Langkah berikutnya akan muncul tampilan untuk konfigurasi phpmyadmin agar dapat terhubung ke *database* MySQL. Maka pilih *YES*.

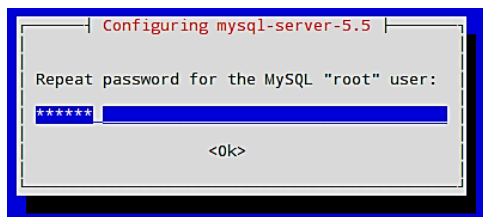


Gambar 3. 9 Konfigurasi PHPMyAdmin untuk membuat koneksi dengan *database*

Tahap berikutnya akan diminta untuk membuat *password* agar dapat masuk ke *database* MySQL secara *online*.

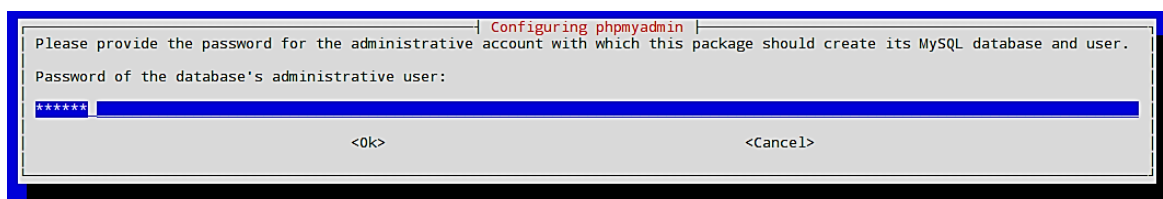


Gambar 3. 10 Membuat *password* user MySQL

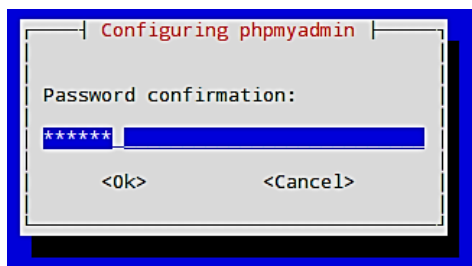


Gambar 3. 11 Repeat password MySQL

Kemudian akan diminta untuk membuat *password* untuk menggunakan *database* phpmyadmin.

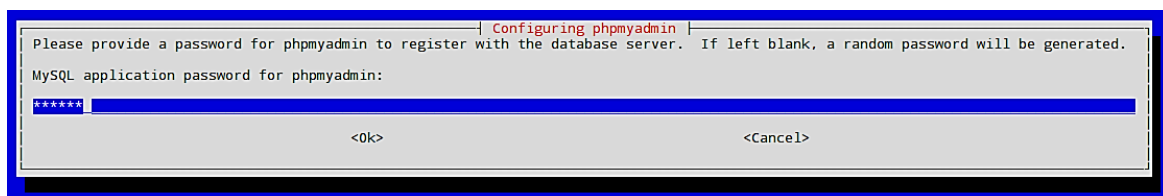


Gambar 3. 12 Membuat *password* user phpmyadmin



Gambar 3. 13 Repeat password phpmyadmin

Setelah itu, diminta untuk membuat *password* untuk aplikasi MySQL untuk phpmyadmin.



Gambar 3. 14 Password aplikasi MySQL untuk phpmyadmin

Untuk langkah berikutnya adalah *setup web server* Apache untuk bisa terhubung ke phpmyadmin dengan mengetikkan *command* sebagai berikut pada terminal.

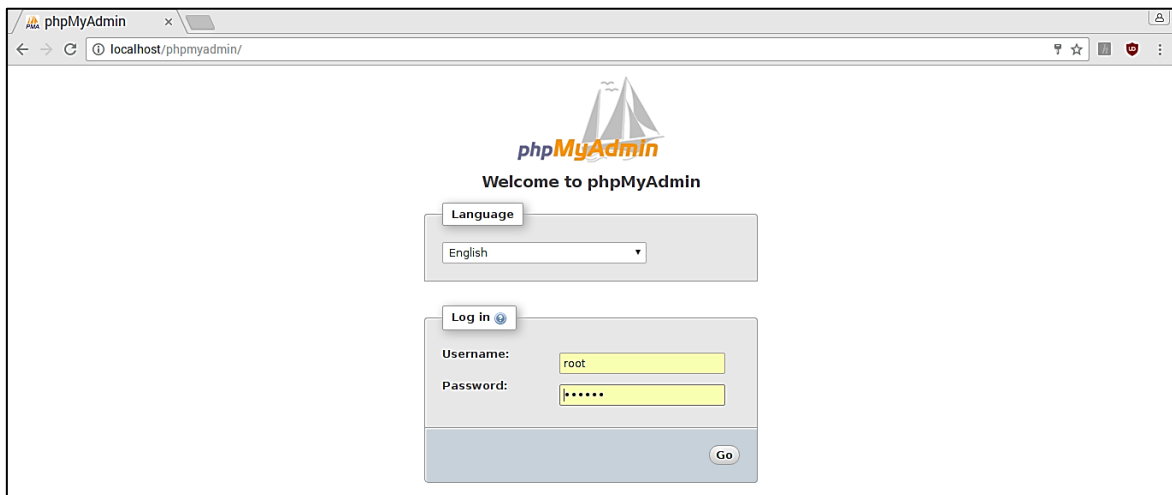
```
pi@raspberrypi:~ $ sudo dpkg-reconfigure -plow phpmyadmin
```

```
pi@raspberrypi:~ $ ln -s /etc/phpmyadmin/apache2.conf /etc/apache2/conf-available/phpmyadmin.conf
```

```
pi@raspberrypi:~ $ a2enconf phpmyadmin
```

```
pi@raspberrypi:~ $ /etc/init.d/apache2 reload
```

Setelah sistem selesai di-*reload* maka dapat dibuka melalui *web browser* dengan menggunakan *IP address* Raspberry Pi dan diikuti “/phpmyadmin” pada akhir alamat pencarian dan akan muncul tampilan seperti gambar 3.15.



Gambar 3. 15 Tampilan login PHPMyAdmin

Pengelolaan *database* MySQL melalui phpmyadmin dapat mempermudah pengguna dalam membuat *database*, menghapus *database*, membuat tabel, menghapus tabel dan lain sebagainya tanpa harus menghafal *query* atau *command* khusus untuk meng-*edit database*.

Perancangan sistem *monitoring smart grid* melalui media sosial dan *website* digunakan tabel dengan struktur sebagai berikut:

Tabel 3. 1
Struktur Tabel *Monitoring Smart Grid*

Nama Kolom	Jenis	Ekstra	Primary Key
ID	Integer	Auto_Increment	√
Tegangan	Decimal (5,2)	-	-
Arus	Decimal (5,2)	-	-
Waktu	Datetime	-	-

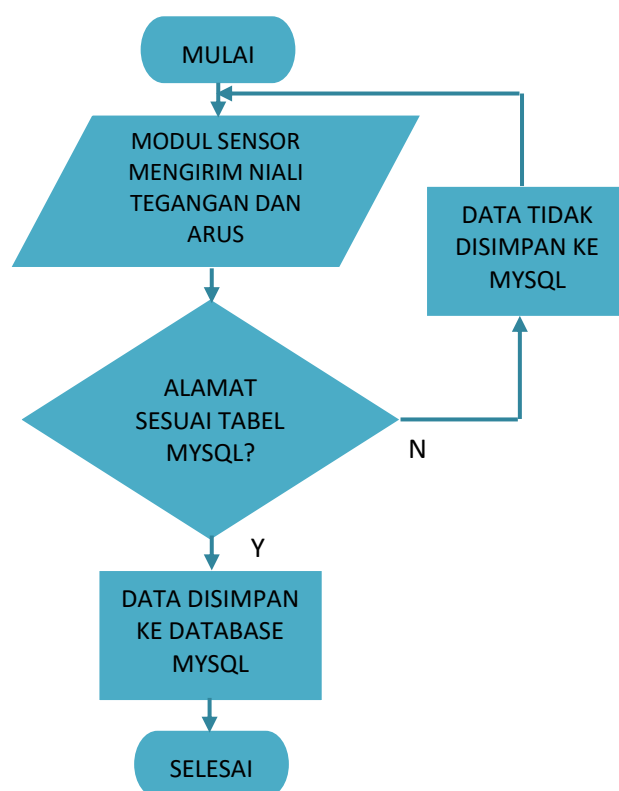
3.2.1.3 Script PHP pada Server

Pada perancangan ini, program PHP digunakan untuk menerima data dari modul sensor pada masing-masing *node* dan menyimpannya ke dalam *database* MySQL. Program PHP

juga digunakan untuk menampilkan nilai-nilai tegangan dan arus pada *smart grid* dalam bentuk grafik.

3.2.1.3.1 Script PHP Penerima Data

Program PHP dibuat sejumlah *node* yang dirancang, sehingga terdapat tiga *file* PHP yang mana masing-masing *file* akan menerima satu paket data dari modul sensor pada *node smart grid*. Data yang diterima akan diteruskan untuk disimpan pada tabel yang sudah disediakan pada *database*. Untuk menampung data dari satu *node* akan disimpan dalam satu tabel, sehingga table yang dibuat sebanyak tiga tabel. Diagram alir penerima data dari modul sensor ditunjukkan dalam gambar 3.16.



Gambar 3. 16 Diagram alir program *server* PHP menyimpan data ke *database* Mysql

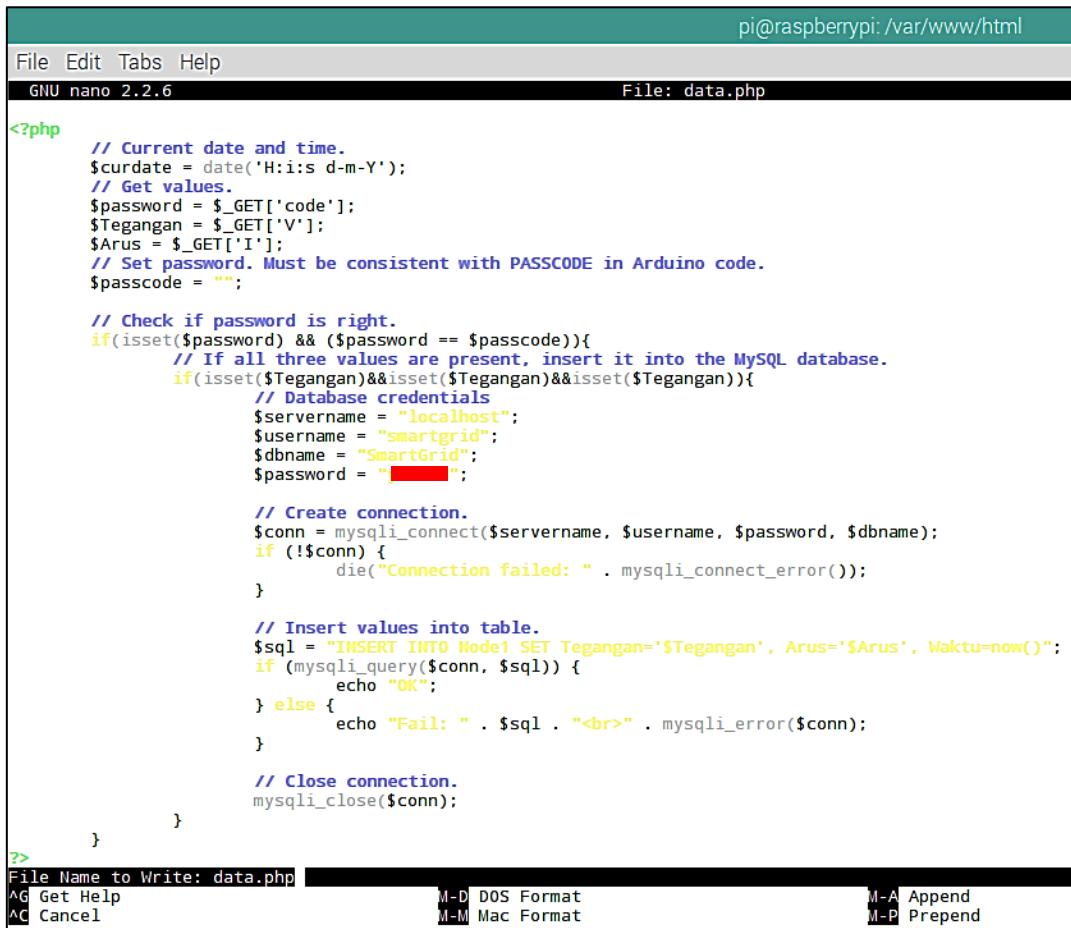
Listing program PHP dapat dilakukan dengan membuat *file* melalui terminal, beri nama *file* PHP tersebut dengan “data.php” dan mengetik *listing* program melalui terminal.

```
pi@raspberrypi:~ $ cd /var/www/html
```

```
pi@raspberrypi:/var/www/html $ sudo nano data.php
```

File PHP yang dibuat berfungsi untuk menyimpan nilai tegangan dan arus yang dikirim oleh modul sensor ke dalam *database*. Pembuatan *file* PHP yaitu sebanyak *node* yang dirancang, sehingga perlu dibuat *file* PHP yang akan menampung data dari masing-masing

node. Masing-masing *file* PHP akan menyimpan data ke dalam tabel yang sudah disediakan didalam *database*. Tahap awal ketika program PHP dijalankan yaitu menerima nilai pembacaan sensor tegangan dan arus melalui internet dengan menggunakan modul Wi-Fi ESP8266. Kemudian data akan cek apakah data yang dikirim *valid* atau tidak. Apabila *valid*, maka data akan disimpan ke dalam tabel pada *database*.



```

pi@raspberrypi: /var/www/html
File Edit Tabs Help
GNU nano 2.2.6 File: data.php

<?php
    // Current date and time.
    $curdate = date('H:i:s d-m-Y');
    // Get values.
    $password = $_GET['code'];
    $Tegangan = $_GET['V'];
    $Arus = $_GET['I'];
    // Set password. Must be consistent with PASSCODE in Arduino code.
    $passcode = "";

    // Check if password is right.
    if(isset($password) && ($password == $passcode)){
        // If all three values are present, insert it into the MySQL database.
        if(isset($Tegangan)&&isset($Tegangan)&&isset($Tegangan)){
            // Database credentials
            $servername = "localhost";
            $username = "smartgrid";
            $dbname = "SmartGrid";
            $password = " ";

            // Create connection.
            $conn = mysqli_connect($servername, $username, $password, $dbname);
            if (!$conn) {
                die("Connection failed: " . mysqli_connect_error());
            }

            // Insert values into table.
            $sql = "INSERT INTO Node1 SET Tegangan='$Tegangan', Arus='$Arus', Waktu=now()";
            if (mysqli_query($conn, $sql)) {
                echo "OK";
            } else {
                echo "Fail: " . $sql . "<br>" . mysqli_error($conn);
            }

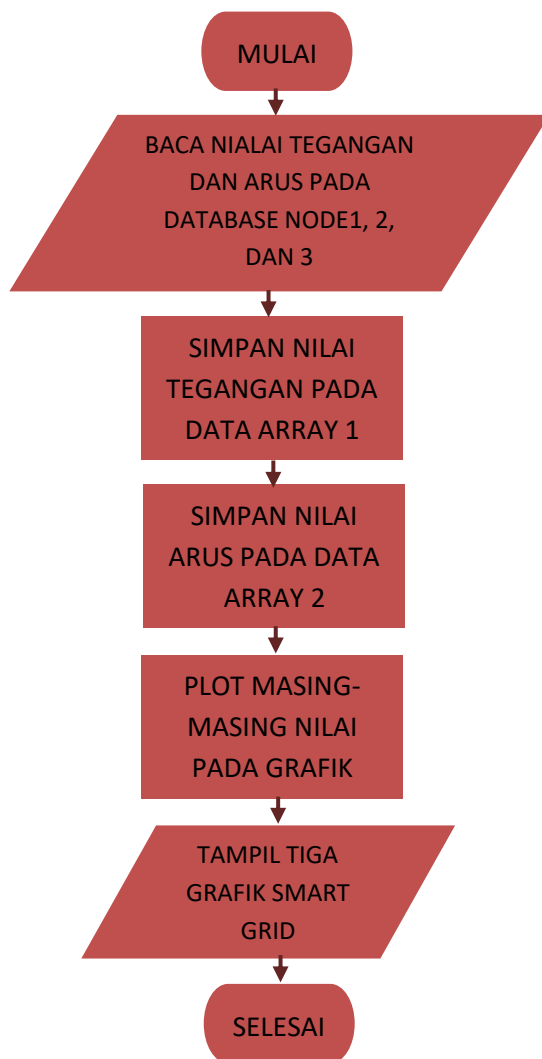
            // Close connection.
            mysqli_close($conn);
        }
    }
}

File Name to Write: data.php
^G Get Help      M-D DOS Format  M-A Append
^C Cancel        M-M Mac Format  M-P Prepend
  
```

Gambar 3. 17 Listing program PHP penyimpan data ke *database*

3.2.1.3.2 Script PHP untuk Menampilkan Grafik

Program PHP untuk menampilkan grafik *monitoring* dibutuhkan tiga *file* PHP untuk membuat tiga grafik dari masing-masing *node*. Sehingga dalam satu laman *web* hanya akan menampilkan satu grafik. Diagram alir penampil grafik ditunjukkan dalam gambar berikut.



Gambar 3. 18 Diagram alir program *server* PHP penampil grafik

Langkah pembuatan *listing* program PHP yang berfungsi untuk menampilkan grafik melalui sebuah *web* yaitu dengan membuat *file* PHP baru dan diberi nama ChartSmartGrid.php. *Listing* program PHP diketik melalui terminal dengan *command* sebagai berikut:

```
pi@raspberrypi:/var/www/html $ sudo nano ChartSmartGrid.php
```

File PHP penampil grafik dibuat sebanyak *node* yang dibuat. Namun cukup satu *file* PHP yang akan menampilkan tiga grafik *node smart grid* masing-masing *node*. Tahap pertama ketika program berjalan adalah membaca nilai tegangan dan arus pada *database*, kemudian nilai tegangan akan disimpan pada variabel *array* 1 dan nilai arus akan disimpan pada variabel *array* 2. Nilai yang telah disimpan pada masing-masing variabel *array* akan di plot ke dalam grafik dengan sumbu Y yang akan menunjukkan besarnya nilai tegangan dan arus dengan satuan V dan A. Sedangkan sumbu X akan menampilkan waktu data tersebut disimpan pada *database*.


```

pi@raspberrypi: /var/www/html
File Edit Tabs Help
GNU nano 2.2.6 File: ChartSmartGrid.php

<?php
mysql_connect('localhost','smartgrid','');
mysql_select_db('SmartGrid');

$sql1 = "SELECT Tegangan, Arus, Waktu FROM Node1";
$query = mysql_query($sql1);
while($data = mysql_fetch_array($query)){
    $V1[] = $data['Tegangan'];
    $I1[] = $data['Arus'];
    $Waktu1[] = $data['Waktu'];
}
$sql2 = "SELECT Tegangan, Arus, Waktu FROM Node2";
$query = mysql_query($sql2);
while($data = mysql_fetch_array($query)){
    $V2[] = $data['Tegangan'];
    $I2[] = $data['Arus'];
    $Waktu2[] = $data['Waktu'];
}
$sql3 = "SELECT Tegangan, Arus, Waktu FROM Node3";
$query = mysql_query($sql3);
while($data = mysql_fetch_array($query)){
    $V3[] = $data['Tegangan'];
    $I3[] = $data['Arus'];
    $Waktu3[] = $data['Waktu'];
}

?>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8">
<title>Monitoring Smart Grid</title>

<script type="text/javascript" src="http://ajax.googleapis.com/ajax/libs/jquery/1.7.1/jquery.min.js"></script>
<script type="text/javascript">

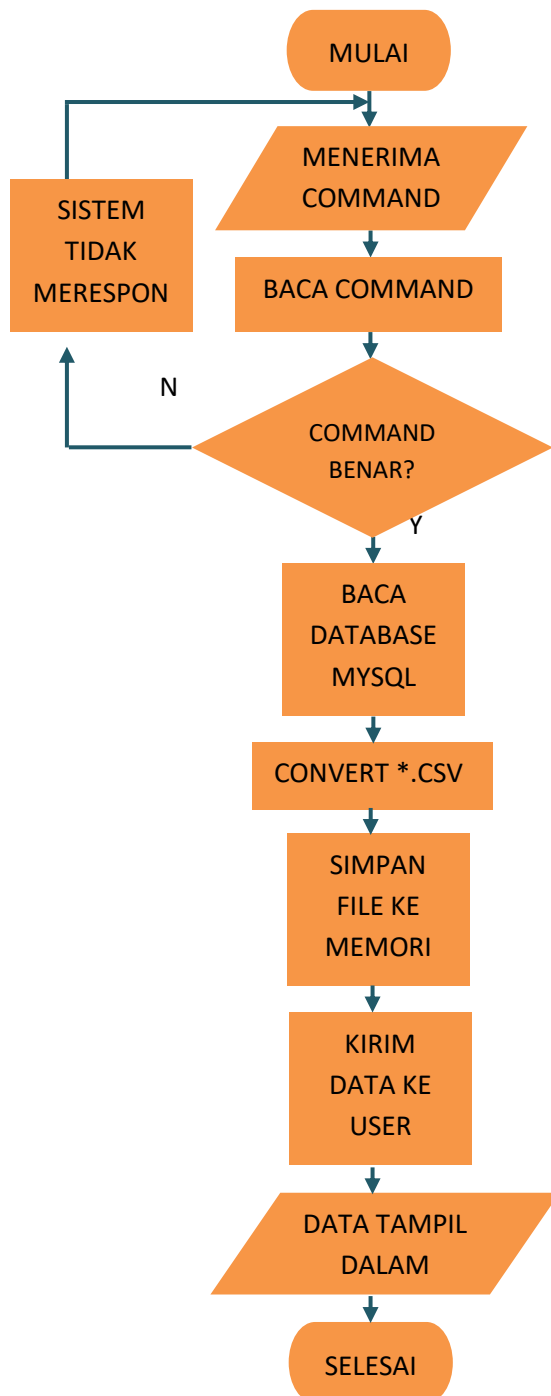
$(function() {
    Highcharts.setOptions({
        chart: {
            backgroundColor: {
AG Get Help      AO WriteOut      AR Read File     AV Prev Page     AK Cut Text      AC Cur Pos
AX Exit          AJ Justify       AW Where Is      AN Next Page     AU UnCut Text    AT To Spell

```

Gambar 3. 19 Listing program PHP penampil grafik

3.2.1.4 Bagian Pengolahan *Command user*

Perancangan sistem menggunakan program Python untuk menjalankan Bot API media sosial, yang berfungsi untuk menerima *command* dari *user* dan mengirimkan informasi tegangan dan arus dalam bentuk *file *.csv*. *Server* Python yang dirancang hanya akan memberikan tanggapan apabila *command* tersebut didefinisikan dalam *listing* program Python. Apabila *command* yang dikirim oleh *user* tidak didefinisikan oleh program Python, maka *server* Python tidak akan memberikan tanggapan. Diagram alir program Python ditunjukkan dalam Gambar 3.20.



Gambar 3. 20 Diagram alir program server Python

Server Python pertama kali akan menunggu *command* yang dikirimkan oleh *user*, ketika ada *command* yang masuk maka *command* akan dibaca, ketika *command* yang diterima server benar, maka akan dilanjutkan ke proses berikutnya. Apabila *command* yang dikirimkan salah, maka server Python tidak akan merespon. Setelah *command* yang dicek benar, maka server akan membaca *database* dan menulis data dari *database* ke dalam *file* *.csv dan menyimpannya ke dalam memori internal Raspberry Pi 3. Setelah proses penulisan *file* *.csv selesai, maka *file* tersebut akan dikirim kepada *user*. Proses pembacaan dan

penulisan data *monitoring smart grid* didasarkan pada rentang waktu yang diinginkan oleh *user*. Apabila waktu *monitoring* yang diminta mempunyai rentang waktu yang pendek, maka lama waktu pembacaan dan penulisan semakin singkat, dan ukuran *file* juga semakin kecil. Begitu pula sebaliknya, apabila rentang waktu yang diminta panjang, maka pembacaan dan penulisan *file* akan lebih lama dan ukuran *file* juga semakin besar.

3.2.1.5 Integrasi Media Sosial

3.2.1.5.1 Integrasi Media Sosial WhatsApp

WhatsApp digunakan sebagai *user* yang akan melakukan proses *switching* dan *monitoring* dari *server*. Pesan dari WhatsApp akan direspon oleh *server* Python dengan cara memberikan respon terhadap *command* yang dikirimkan. Konfigurasi Bot API WhatsApp dilakukan melalui yowsup yang sudah disediakan oleh *developer* media sosial WhatsApp. Berikut adalah tahap yang dilakukan untuk menginstall yowsup.

```
pi@raspberrypi:~ $ sudo rpi-update
```

```
pi@raspberrypi:~ $ sudo apt-get install python-dateutil
```

```
pi@raspberrypi:~ $ sudo apt-get install python-setuptools
```

```
pi@raspberrypi:~ $ sudo apt-get install libevent-dev
```

```
pi@raspberrypi:~ $ sudo apt-get install python-dev
```

```
pi@raspberrypi:~ $ sudo apt-get install ncurses-dev
```

```
pi@raspberrypi:~ $ git clone git://github.com/tgalal/yowsup.git
```

```
pi@raspberrypi:~ $ cd yowsup
```

```
pi@raspberrypi:/yowsup $ sudo python setup.py install
```

Setelah proses *install* telah dilakukan, selanjutnya adalah melakukan registrasi *phone number* untuk mendapatkan code verifikasi. Dengan keterangan, cc adalah kode negara, mcc adalah, mnc adalah.

```
pi@raspberrypi:/yowsup $ python yowsup-cli registration --requestcode sms
--phone 62877xxxxxxx --cc 62 --mcc 510 --mnc 11
```

```
yowsup-cli v2.0.15
yowsup v2.5.2

Copyright (c) 2012-2016 Tarek Galal
http://www.openwhatsapp.org

This software is provided free of charge. Copying and redistribution is
encouraged.

If you appreciate this software and you would like to support future
development please consider donating:
http://openwhatsapp.org/yowsup/donate

INFO:yowsup.common.http.warequest:{"login":"62877[REDACTED]","status":"sent","length":6,"method":"sms","retry_after":65,"sms_wait":65,"voice_wait":65}

status: sent
retry_after: 65
length: 6
login: 62877[REDACTED]
method: sms
```

Gambar 3. 21 Tampilan yowsup berhasil mengirimkan kode verifikasi berupa pesan sms

Tahap selanjutnya adalah memasukkan kode registrasi untuk mendapatkan *password* akun yang akan digunakan sebagai *user*.

```
pi@raspberrypi:/yowsup $ python yowsup-cli registration --register 983-
xxx --phone 62877xxxxxxx --cc 62
```

```
yowsup-cli v2.0.15
yowsup v2.5.2

Copyright (c) 2012-2016 Tarek Galal
http://www.openwhatsapp.org

This software is provided free of charge. Copying and redistribution is
encouraged.

If you appreciate this software and you would like to support future
development please consider donating:
http://openwhatsapp.org/yowsup/donate

INFO:yowsup.common.http.warequest:{"status":"ok","login":"62877[REDACTED]","type":"existing","pw":"QqKqM[REDACTED]","expiration":444444444.0,"kind":"free","price":"$0.99","cost":"0.99","currency":"USD","price_expiration":1511855182}

status: ok
kind: free
pw: QqKqM[REDACTED]
price: $0.99
price_expiration: 1511855182
currency: USD
cost: 0.99
expiration: 444444444.0
login: 62877[REDACTED]
type: existing
```

Gambar 3. 22 Tampilan pengiriman password akun WhatsApp

Setelah proses berhasil dan *password* diterima, selanjutnya adalah memindahkan *password* pada yowsup-cli.

```

yowsup-cli
File Edit Search Options Help
#!/usr/bin/env python
__version__ = "2.0.15"
__author__ = "Tarek Galal"

import sys, argparse, yowsup, logging
from yowsup.env import YowsupEnv

HELP_CONFIG = """
##### Yowsup Configuration Sample #####
#
# =====
# The file contains info about your WhatsApp account. This is used during registration and login.
# You can define or override all fields in the command line args as well.
#
# Country code. See http://www.ipipi.com/help/telephone-country-codes.htm. This is now required.
cc=62
#
# Your full phone number including the country code you defined in 'cc', without preceding '+' or '00'
phone=62877[REDACTED]
#
# You obtain this password when you register using Yowsup.
password=QQaKQM[REDACTED]
#####
"""

CR_TEXT = """yowsup-cli v{cliVersion}
yowsup v{yowsupVersion}

Copyright (c) 2012-2016 Tarek Galal
http://www.openwhatsapp.org

This software is provided free of charge. Copying and redistribution is
encouraged.

```

Gambar 3. 23 Konfigurasi akun yowsup

Kemudian lakukan konfigurasi dengan mengetikkan *command* berikut pada *line terminal*.

```
pi@raspberrypi:/yowsup $ yowsup-cli demos --yowsup --config config
```

```

Traceback (most recent call last):
  File "/usr/local/bin/yowsup-cli", line 4, in <module>
    __import__('pkg_resources').run_script('yowsup2==2.5.2', 'yowsup-cli')
  File "/usr/lib/python2.7/dist-packages/pkg_resources.py", line 2876, in <module>
    working_set = WorkingSet._build_master()
  File "/usr/lib/python2.7/dist-packages/pkg_resources.py", line 451, in _build_master
    return cls._build_from_requirements(__requires__)
  File "/usr/lib/python2.7/dist-packages/pkg_resources.py", line 464, in _build_from_requirements
    dists = ws.resolve(reqs, Environment())
  File "/usr/lib/python2.7/dist-packages/pkg_resources.py", line 644, in resolve
    raise VersionConflict(dist, req)
pkg_resources.VersionConflict: (six 1.8.0 (/usr/lib/python2.7/dist-packages), Requirement.parse('six>=1.9'))

```

Gambar 3. 24 Permasalahan versi python-pip six 1.8

Namun proses tidak berjalan dengan baik, yang terlihat adalah permasalahan dalam versi python-pip. Versi yang sedang digunakan adalah *six* 1.8.0, sedangkan syarat versi minimum adalah *six* 1.9. Maka tahap selanjutnya adalah melakukan *upgrade* python-pip kedalam versi *six* 1.9. Berikut adalah tahap yang dapat digunakan untuk melakukan *upgrade* versi python-pip.

```
pi@raspberrypi:~ $ sudo apt-get remove python-pip
```

```
pi@raspberrypi:~ $ wget https://raw.githubusercontent.com/pypa/pip/master/contrib/get-pip.py
```

```
pi@raspberrypi:~ $ wget https://bootstrap.pypa.io/get-pip.py
```

```
pi@raspberrypi:~ $ python get-pip.py
```

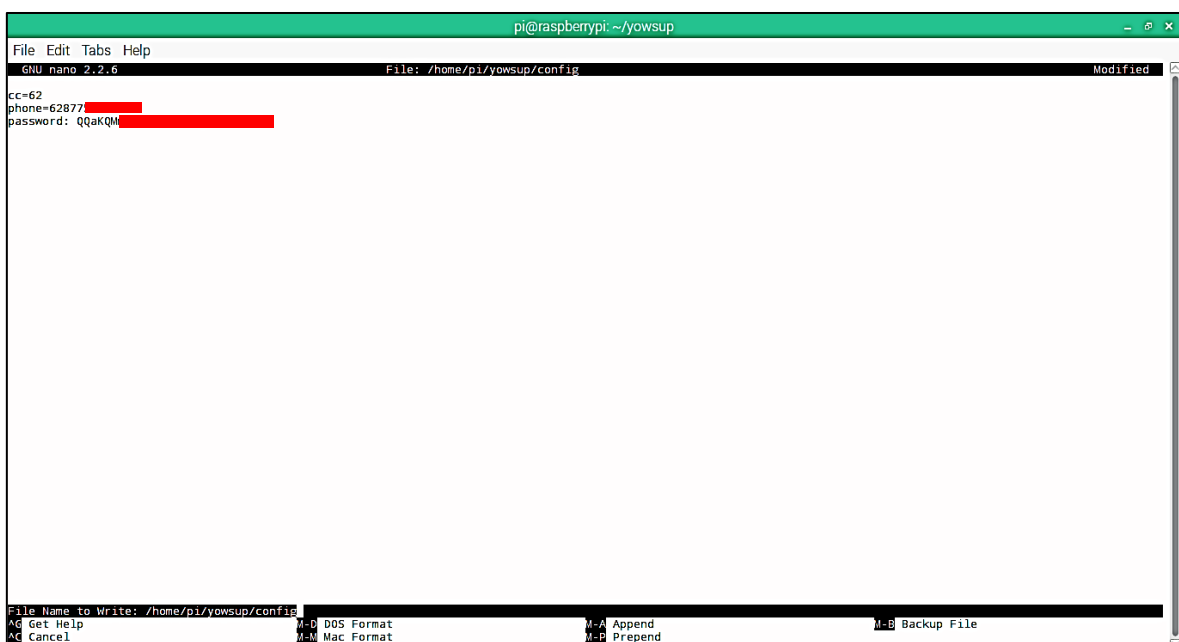
```
pi@raspberrypi:~ $ sudo apt-get install python-pip
```

```
pi@raspberrypi:~ $ sudo pip install --upgrade pip
```

```
pi@raspberrypi:~ $ sudo pip install six
```

```
pi@raspberrypi:~ $ sudo pip install six==1.90
```

Setelah proses install python-pip six 1.9 telah berhasil, selanjutnya adalah kembali mengulang proses registrasi *phone number*. Setelah memperoleh *password*, masukkan *password* ke dalam program konfigurasi seperti pada gambar dibawah.



Gambar 3. 25 Konfigurasi akun yowsup

Setelah program disimpan, lakukan kembali proses konfigurasi dengan *command* berikut.

```
pi@raspberrypi:/yowsup $ yowsup-cli demos --yowsup --config config
```

```

Traceback (most recent call last):
  File "/usr/local/bin/yowsup-cli", line 4, in <module>
    import ('pkg_resources').run_script('yowsup2==2.5.2', 'yowsup-cli')
  File "/usr/lib/python2.7/dist-packages/pkg_resources.py", line 534, in run_script
    self.require(requires)[0].run_script(script_name, ns)
  File "/usr/lib/python2.7/dist-packages/pkg_resources.py", line 1438, in run_script
    execfile(script_filename, namespace, namespace)
  File "/usr/local/lib/python2.7/dist-packages/yowsup2-2.5.2-py2.7.egg/EGG-INF0/scripts/yowsup-cli", line 368, in <module>
    if not parser.process():
  File "/usr/local/lib/python2.7/dist-packages/yowsup2-2.5.2-py2.7.egg/EGG-INF0/scripts/yowsup-cli", line 268, in process
    self.startCmdline()
  File "/usr/local/lib/python2.7/dist-packages/yowsup2-2.5.2-py2.7.egg/EGG-INF0/scripts/yowsup-cli", line 291, in startCmdline
    credentials = self._getCredentials()
  File "/usr/local/lib/python2.7/dist-packages/yowsup2-2.5.2-py2.7.egg/EGG-INF0/scripts/yowsup-cli", line 284, in _getCredentials
    assert "password" in config and "phone" in config, "Must specify at least phone number and password in config file"
AssertionError: Must specify at least phone number and password in config file

```

Gambar 3. 26 Tampilan error konfigurasi

Terlihat bahwa *error* tetap terjadi. Namun untuk kali ini error berasal dari password dan *phone number* yang tidak ditentukan pada proses konfigurasi. Peneliti telah mencoba beberapa cara untuk memulihkan error tersebut. Namun hasilnya tetap saja, bahkan semakin banyak keterangan error yang tampil.

Berikut adalah *command* pada *line terminal* yang dapat diketikkan untuk menampilkan detail akun.

```
pi@raspberrypi:/yowsup $ yowsup-cli demos --yowsup --config mydetail
```

```

yowsup-cli v2.0.15
yowsup v2.5.2

Copyright (c) 2012-2016 Tarek Galal
http://www.openwhatsapp.org

This software is provided free of charge. Copying and redistribution is
encouraged.

If you appreciate this software and you would like to support future
development please consider donating:
http://openwhatsapp.org/yowsup/donate

Yowsup Cli client
=====
Type /help for available commands

[offline]:█

```

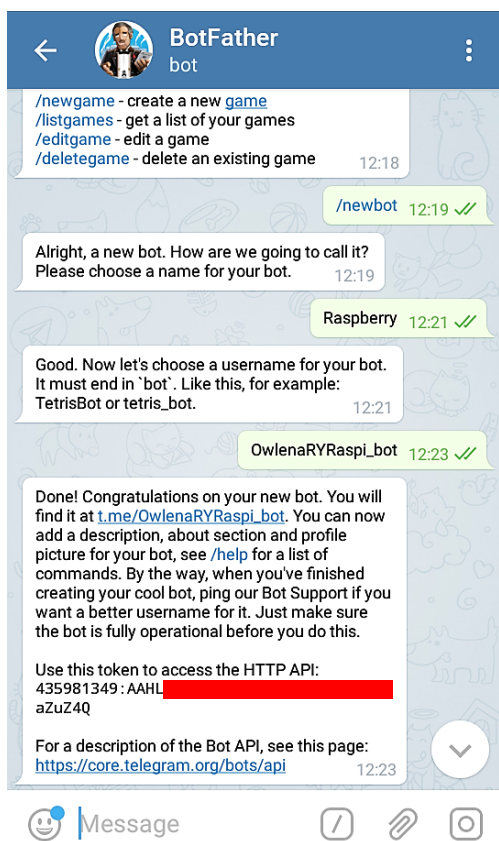
Gambar 3. 27 Tampilan detail akun yowsup WhatsApp

Terlihat bahwa akun akan selalu dikatakan *offline*, hal tersebut karena WhatsApp tidak menerima konfigurasi yang telah dilakukan. Banyak pengguna lain yang sudah beralih dari penggunaan Bot WhatsApp, dikarenakan sistem keamanan yang sangat ketat pada WhatsApp. Sering kali WhatsApp melakukan pemblokiran akun karena dianggap bukan *official account* WhatsApp.

3.2.1.5.2 Integrasi Media Sosial Telegram

Dalam perancangan sistem, Bot API Telegram digunakan sebagai akun yang akan melakukan proses *switching* dan *monitoring* dari *server* ke *user* melalui media sosial Telegram. Bot Api Telegram akan dijalankan oleh *server* Python dengan cara memberikan

respon terhadap *command* yang dikirimkan oleh *user*. Konfigurasi Bot API Telegram dilakukan melalui BotFather yang sudah disediakan oleh *developer* media sosial Telegram. Langkah pertama untuk membuat Bot API Telegram yaitu dengan mengetikkan *command* “/newbot”. Langkah berikutnya yaitu memberikan nama Bot yang ingin dibuat, dalam kasus ini penulis memberikan nama “Raspberry”. Tahap berikutnya yaitu memberikan *username* untuk Bot yang ingin dibuat, yaitu “OwlenaRYRaspi_bot”. Proses terakhir yaitu BotFather akan mengirimkan token yang akan digunakan sebagai *server* Python. Proses pembuatan Bot API Telegram seperti yang ditunjukkan dalam gambar 3.28.



Gambar 3. 28 Konfigurasi Bot API Telegram

Bot API Telegram harus diidentifikasi oleh Raspberry Pi melalui *script* Python agar bot yang sudah dibuat dapat dikenali oleh Raspberry. Proses install dilakukan dengan mengikuti langkah-langkah seperti berikut:

```
pi@raspberrypi:~ $ sudo apt-get install python-pip
```

```
pi@raspberrypi:~ $ sudo pip install telepot
```

Setelah proses install selesai, selanjutnya adalah proses identifikasi Bot API Telegram sesuai dengan token yang didapatkan dari BotFather.


```
pi@raspberrypi:~ $ python
```

```
Python 2.7.9 (default, Sep 17 2016, 20:26:04)
[GCC 4.9.2] on linux2
Type "help", "copyright", "credits" or "license" for more information.
>>> import telepot
>>> bot = telepot.Bot('435981349:AAHL[REDACTED]')
>>> bot.getMe()
{'username': u'0wlenaRYRaspi_bot', u'first_name': u'Raspberry', u'is_bot': True, u'id': 435981349}
>>> exit()
```

Gambar 3. 29 Identifikasi Bot API Telegram

Saat Bot telah teridentifikasi, maka Bot API Telegram telah dapat digunakan untuk berbagai pengolahan berbagai pesan.

3.2.1.6 Integrasi Google Drive

Google Drive telah banyak dimanfaatkan sebagai tempat penyimpanan data yang aman, sehingga penggunaannya semakin populer. Pada penelitian ini, Google Drive dimanfaatkan untuk tempat penyimpanan *database* yang dapat diakses dimanapun. Berikut adalah perintah *install* Google Drive.

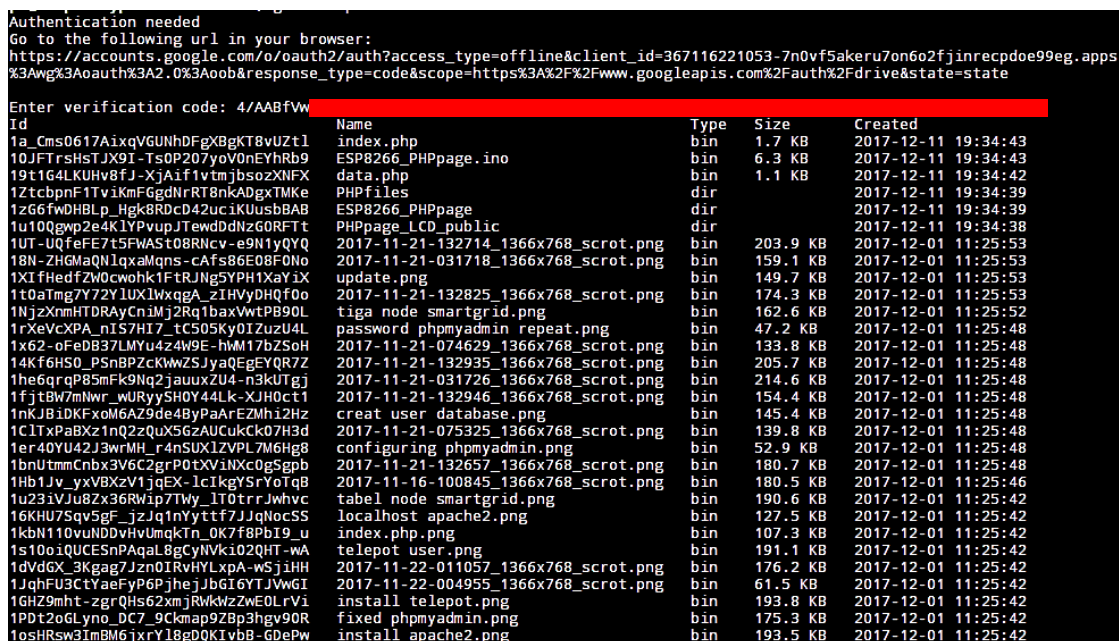
```
pi@raspberrypi:~ $ wget https://docs.google.com/uc?id=0B3X9G1R6EmbnVXNL
anp4ZFRRbzbzg&export=download
```

```
pi@raspberrypi:~ $ chmod +x gdrive-linux-rpi
```

```
pi@raspberrypi:~ $ sudo install gdrive-linux-rpi /usr/local/gdrive-rpi
```

Setelah proses *install* berhasil, tulis *command* berikut untuk memunculkan URL untuk otentikasi. Setelah dibuka pada *browser* maka akan mendapatkan kode verifikasi.

```
pi@raspberrypi:~ $ grdrive-rpi list
```



Gambar 3. 30 Otentikasi Google Drive



Gambar 3. 31 Kode verifikasi Google Drive

Otentikasi berhasil dilakukan, *server* dapat melakukan pengunduhan *database* pada Google Drive dengan tampilan Google Spreadsheet.

3.3 Pengujian Alat

Pengujian alat dilakukan pada setiap bagian blok diagram untuk mengetahui hasil perancangan. Pengujian yang dilakukan meliputi:

3.3.1 Pengujian Server Raspberry Pi 3

3.3.1.1 Pengujian RAM Raspberry Pi 3

Pengujian ini bertujuan mengetahui beban RAM pada Raspberry Pi 3 untuk bekerja sebagai *server*. Pengujian ini dilakukan dengan menjalankan *server* sesuai dengan kinerja pada perancangan. RAM yang digunakan oleh Raspberry Pi 3 dapat diketahui dengan cara mengetikkan *command* sebagai berikut pada terminal:

```
pi@raspberrypi:~$ free -m
```

	total	used	free	shared	buffers	cached
Mem:	923	854	68	104	29	365
-/+ buffers/cache:		459	463			
Swap:	99	5	94			

Gambar 3. 32 Penggunaan RAM Raspberry Pi 3

3.3.1.2 Pengujian Suhu CPU Raspberry Pi 3

Pengujian ketahanan Raspberry Pi ini dimaksudkan untuk menguji ketahanan dari CPU Raspberry Pi dalam hal membaca suhu dari CPU Raspberry Pi 3 yang digunakan sebagai *server* untuk jangka waktu yang lama. Suhu CPU Raspberry Pi 3 dapat diketahui dengan cara mengetikkan *command* sebagai berikut pad terminal:

```
pi@raspberrypi:~ $ vcgencmd measure_temp
```

```
|temp=53.7'C
```

Gambar 3. 33 Suhu CPU Raspberry Pi 3

3.3.2 Pengujian Perangkat Lunak pada Server

3.3.2.1 Pengujian Pengiriman Data ke Database MySQL

Pengujian ini dilakukan dengan cara menghubungkan modul sensor ke internet dan mengirimkan data pembacaan sensor ke *database* MySQL. Pengujian ini bertujuan untuk mengetahui apakah data yang dikirim oleh modul sensor dapat tersimpan pada database MySQL dan sesuai dengan format tabel yang telah dibuat.

3.3.2.2 Pengujian Kecepatan Pengiriman Data ke Database MySQL

Pengujian ini dilakukan untuk mengetahui lama waktu pengiriman data dari modul sensor *smart grid* ke *database* MySQL. Pengujian dilakukan dengan memberikan *delay* yang berbeda-beda pada modul sensor untuk mengirim data ke *database* MySQL.

3.3.2.3 Pengujian Pengolahan Database MySQL

Pengujian ini dilakukan dengan cara data pengiriman dari modul sensor yang disimpan pada database MySQL dapat diolah oleh *server* dan dapat di akses oleh *user* melalui media sosial. Pengujian ini dilakukan untuk mengetahui apakah pengolahan *database* yang dibuat dapat berjalan dengan baik atau tidak.

3.3.2.4 Pengujian Menampilkan Data pada Grafik

Pengujian ini dilakukan dengan cara data yang disimpan pada *database* dapat ditampilkan dalam sebuah grafik yang dapat dibuka melalui *website*. Pengujian ini dilakukan untuk mengetahui apakah grafik yang ditampilkan secara *real time* atau tidak.

3.3.2.5 Pengujian Pengiriman Command dari Media Sosial

Pengujian ini dilakukan dengan cara *user* mengirim pesan atau *command* dari media sosial ke *server*. Pengujian bertujuan untuk mengetahui apakah *server* dapat menerima dan merespon *command* atau tidak.

3.3.2.6 Pengujian Pengiriman Data ke User

Pengujian ini dilakukan dengan cara data yang diolah oleh *server* akan di kirim ke *user* dalam format *file* tertentu ketika *user* mengirimkan *command* yang sesuai. Pengujian ini

bertujuan untuk mengetahui apakah *server* dapat mengirim data ke *user* dengan baik atau tidak. Serta untuk mengetahui apakah *server* mampu membedakan dan merespon *command* permintaan data *monitoring* atau proses *switching* sistem.

3.3.2.7 Pengujian Ukuran Data pada Database

Pengujian dilakukan dengan mengisi *database* dengan informasi nilai tegangan, arus dan waktu dengan banyaknya jumlah baris yang ditentukan. Pengujian bertujuan untuk mengetahui kapasitas data yang tersimpan pada *database*.

3.3.2.8 Pengujian Penyimpanan Database pada Google Drive

Pengujian dilakukan dengan mengunggah data yang telah tersimpan dalam format *.csv pada *server* ke Google Drive. Pengujian bertujuan untuk mengetahui keberhasilan pengunggahan data serta waktu yang dibutuhkan untuk mengunggah data dengan kapasitas berbeda-beda.

3.3.2.9 Pengujian Command Switching

Pengujian dilakukan dengan mengirimkan *command* berupa perintah *switching ON* atau *OFF* sistem. Pengujian dilakukan untuk mengetahui apakah sistem akan menjalankan perintah *switching* tersebut dengan benar atau tidak, serta untuk mengetahui lama waktu dari pengiriman *command* hingga sistem merespon.

3.3.2.10 Pengujian Menggunakan Dua User

Pengujian dilakukan dengan melakukan proses pengiriman *command* secara bersamaan pada dua *user*. Tujuannya untuk mengetahui respon yang diterima masing-masing *user* saat permintaan dilakukan serentak serta untuk mengetahui lama waktu penerimaan masing-masing *user*.

3.3.3 Pengujian Keseluruhan

Pengujian keseluruhan sistem dilakukan dengan menjalankan sistem. Parameter yang diuji adalah pengiriman data pembacaan sensor ke *database* MySQL, pengolahan data untuk ditampilkan pada grafik pada laman *web*, pengiriman *command* dari *user* ke *server*, penerimaan *command* dari *user* pada *server*, pengiriman respon dari *server* ke *user* sesuai dengan *command* yang dikirim dari *user*, pengunggahan data pada Google Drive, pengiriman *command switching* penggunaan dua *user* bersamaan.

BAB IV

HASIL DAN PEMBAHASAN

Pembahasan hasil pengujian sistem *switching* dan *monitoring smart grid* dilakukan untuk menganalisis sistem yang telah dirancang dan yang telah dibuat apakah telah bekerja sesuai perancangan. Pengujian dilakukan pada tiap-tiap blok dengan tujuan untuk mengamati apakah tiap blok sistem sudah sesuai dengan perancangan, kemudian dilanjutkan dengan pengujian secara keseluruhan sistem. Adapun pengujian yang perlu dilakukan sebagai berikut:

1. Pengujian *Server Raspberry Pi*
 - a) Pengujian RAM Raspberry Pi 3
 - b) Pengujian suhu CPU Raspberry Pi 3
2. Pengujian Perangkat Lunak *Server*
 - a) Pengujian pengiriman data ke *database*
 - b) Pengujian kecepatan pengiriman data ke *database*
 - c) Pengujian pengolahan *database*
 - d) Pengujian menampilkan grafik
 - e) Pengujian pengiriman *command* dari media sosial Telegram
 - f) Pengujian pengiriman *file *.csv*
 - g) Pengujian ukuran *file *.csv* dan lama waktu pengiriman
 - h) Pengujian pengiriman data ke Google Drive
 - i) Pengujian dua *user*
 - j) Pengujian *switching*
3. Pengujian keseluruhan

4.1 Pengujian Server Raspberry Pi 3

4.1.1 Pengujian RAM Raspberry Pi 3

Pengujian dilakukan dengan menjalankan Raspberry Pi 3 sebagai *server* dan melakukan pemantauan penggunaan RAM. Kondisi RAM sangat berpengaruh pada kecepatan respon *server* dalam memproses sistem. Pengujian RAM dilakukan sebanyak 5 kali. Hasil pengujian ditunjukkan dalam tabel 4.1.

Tabel 4. 1
Hasil pengujian RAM Raspberry Pi 3

Pengujian ke-	RAM (MB)
1	674
2	880
3	854
4	885
5	804
Rata-rata	819,4

Pada tabel 4.1 dapat disimpulkan bahwa rata-rata penggunaan RAM Raspberry yang dilakukan selama 5 hari adalah 819,4.

4.1.2 Pengujian Suhu CPU Raspberry Pi 3

Pengujian suhu CPU Raspberry Pi 3 dilakukan dengan cara menjalankan *server* selama 5 hari dan mengamati suhu CPU Raspberry Pi 3 sebanyak 1 kali per hari. Hasil pengujian ditunjukkan dalam tabel 4.2.

Tabel 4. 2
Hasil pengujian suhu CPU Raspberry Pi 3

Pengujian ke-	Suhu (°C)
1	54,2
2	55,8
3	53,7
4	54,2
5	54,8
Rata-rata	54,5

Pada tabel 4.2 dapat disimpulkan bahwa rata-rata kondisi suhu pada CPU Raspberry selama 5 hari adalah 54,5 °C.

4.2 Pengujian Perangkat Lunak Server

4.2.1 Pengujian Data Pengiriman Sensor

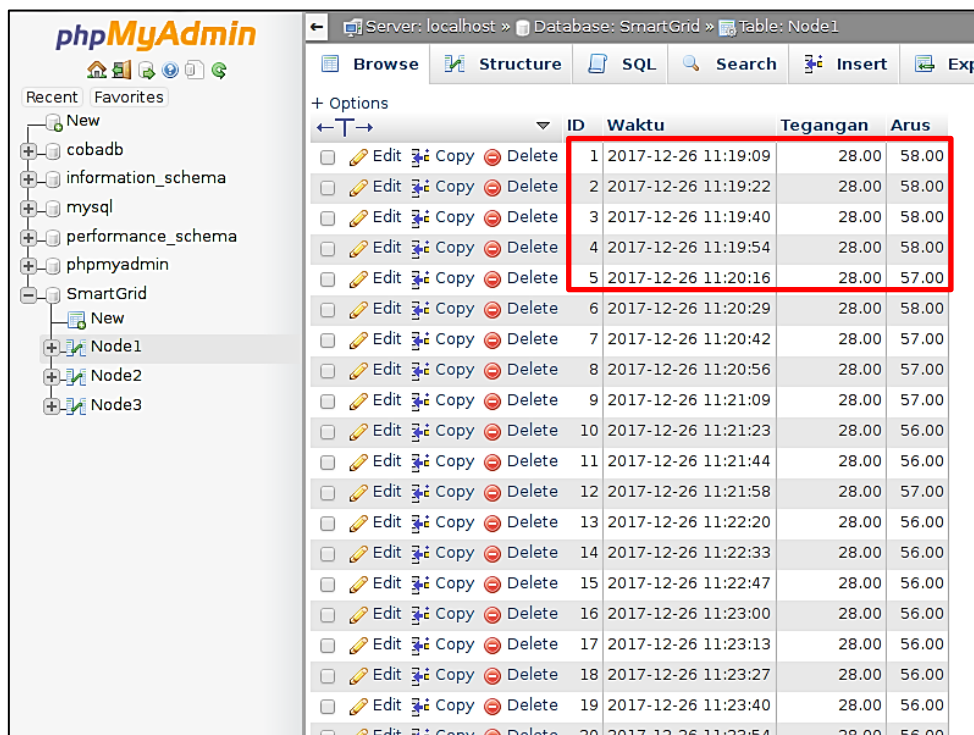
Pengujian dilakukan menggunakan sensor DHT, Arduino dan modul ESP8266. Tidak menggunakan sensor tegangan dan arus karena tujuan pengujian adalah hanya untuk mengetahui data yang diterima *database* MySQL apakah sama dengan yang dikirim oleh modul sensor, pengiriman dilakukan melalui pemrograman PHP.

Tabel 4. 3

Data pengiriman modul sensor DHT node 1 ke database

Data Sensor		Database	
Temperature (C)	Humadity (%)	Tegangan (V)	Arus (A)
28.00	58	28.00	58.00
28.00	58	28.00	58.00
28.00	58	28.00	58.00
28.00	58	28.00	58.00
28.00	57	28.00	57.00

Pengujian pengiriman data dilakukan pada *node1*, pada pengujian ini data dapat terkirim ke *database* yang sudah disediakan melalui *web* dengan menentukan *ip address (local)* Rapberry Pi 3. Berdasarkan tabel 4.3 dapat disimpulkan bahwa data yang dikirim oleh sensor dapat disimpan dengan tepat pada *database*.



The screenshot shows the phpMyAdmin interface. On the left, the database structure is visible, including 'SmartGrid' and its tables 'Node1', 'Node2', and 'Node3'. The main area displays the 'Node1' table with the following data:

ID	Waktu	Tegangan	Arus
1	2017-12-26 11:19:09	28.00	58.00
2	2017-12-26 11:19:22	28.00	58.00
3	2017-12-26 11:19:40	28.00	58.00
4	2017-12-26 11:19:54	28.00	58.00
5	2017-12-26 11:20:16	28.00	57.00
6	2017-12-26 11:20:29	28.00	58.00
7	2017-12-26 11:20:42	28.00	57.00
8	2017-12-26 11:20:56	28.00	57.00
9	2017-12-26 11:21:09	28.00	57.00
10	2017-12-26 11:21:23	28.00	56.00
11	2017-12-26 11:21:44	28.00	56.00
12	2017-12-26 11:21:58	28.00	57.00
13	2017-12-26 11:22:20	28.00	56.00
14	2017-12-26 11:22:33	28.00	56.00
15	2017-12-26 11:22:47	28.00	56.00
16	2017-12-26 11:23:00	28.00	56.00
17	2017-12-26 11:23:13	28.00	56.00
18	2017-12-26 11:23:27	28.00	56.00
19	2017-12-26 11:23:40	28.00	56.00
20	2017-12-26 11:23:54	28.00	56.00

Gambar 4. 1 Tampilan database PHPMyAdmin

Dapat dilihat bahwa data dari modul sensor diterima dan disimpan dengan tepat oleh database phpMyAdmin.

4.2.2 Pengujian Kecepatan Pengiriman ke Database

Pengujian ini dilakukan untuk mengetahui lama waktu pengiriman data dari modul sensor smart grid ke database MySQL. Pengujian dilakukan dengan memberikan *delay* yang berbeda-beda pada modul sensor untuk mengirim data ke database MySQL. Hasil pengujian kecepatan pengiriman ditunjukkan dalam tabel 4.4.

Tabel 4. 4
Kecepatan pengiriman data ke database

Delay Modul Sensor (ms)	Rata-Rata (s)
500	4,11
1000	4,33
2000	5,89
5000	7,55
10000	14,33

Pengujian kecepatan pengiriman dilakukan dengan memberikan *delay* (waktu tunda) yang berbeda-beda pada modul sensor untuk mengirim data ke *database* MySQL. Berdasarkan tabel 4.4 didapatkan *delay* yang paling mendekati delay asli adalah 10 detik dengan rata-rata 14,33 detik.

4.2.3 Pengujian Pengolahan *Database* MySQL

Pengujian ini dilakukan untuk mengetahui data yang tersimpan pada *database* apakah dapat dibaca melalui program Python sesuai dengan waktu yang diinginkan. Berikut adalah tampilan hasil program Python membaca database.

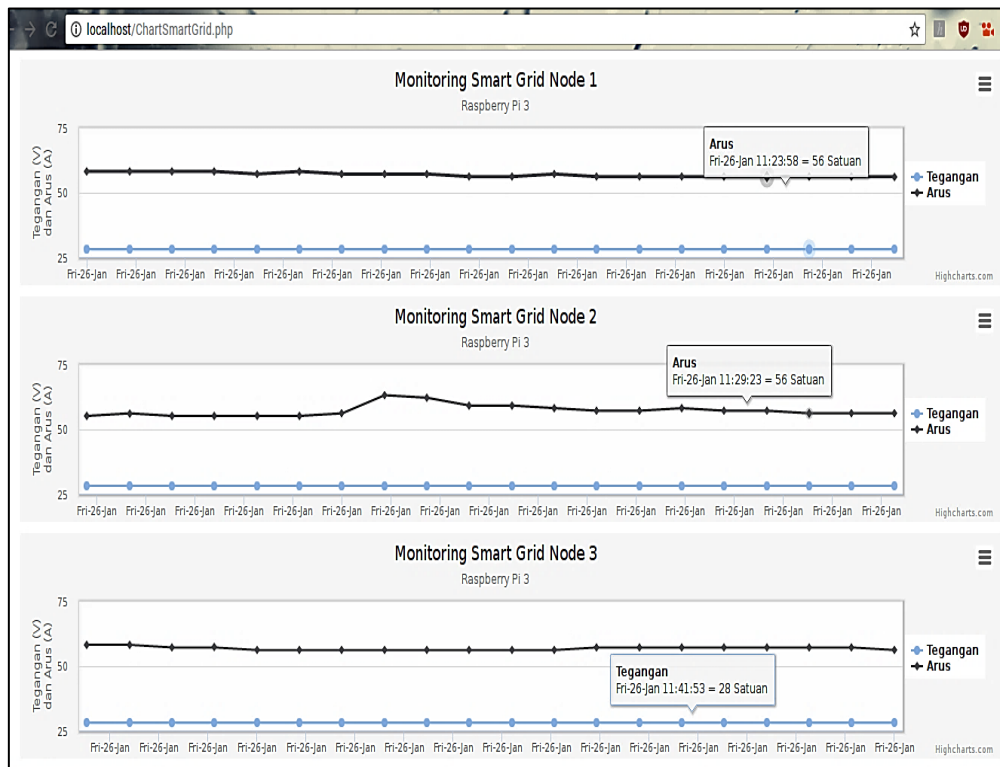
```
pi@raspberrypi:~/Data $ python Read_Mysql.py
Database Version : 5.5.57-0+deb8u1
(1L, datetime.datetime(2017, 12, 26, 11, 19, 9), Decimal('28.00'), Decimal('58.00'))
(2L, datetime.datetime(2017, 12, 26, 11, 19, 22), Decimal('28.00'), Decimal('58.00'))
(3L, datetime.datetime(2017, 12, 26, 11, 19, 40), Decimal('28.00'), Decimal('58.00'))
(4L, datetime.datetime(2017, 12, 26, 11, 19, 54), Decimal('28.00'), Decimal('58.00'))
(5L, datetime.datetime(2017, 12, 26, 11, 20, 16), Decimal('28.00'), Decimal('57.00'))
(6L, datetime.datetime(2017, 12, 26, 11, 20, 29), Decimal('28.00'), Decimal('58.00'))
(7L, datetime.datetime(2017, 12, 26, 11, 20, 42), Decimal('28.00'), Decimal('57.00'))
(8L, datetime.datetime(2017, 12, 26, 11, 20, 56), Decimal('28.00'), Decimal('57.00'))
(9L, datetime.datetime(2017, 12, 26, 11, 21, 9), Decimal('28.00'), Decimal('57.00'))
(10L, datetime.datetime(2017, 12, 26, 11, 21, 23), Decimal('28.00'), Decimal('56.00'))
(11L, datetime.datetime(2017, 12, 26, 11, 21, 44), Decimal('28.00'), Decimal('56.00'))
(12L, datetime.datetime(2017, 12, 26, 11, 21, 58), Decimal('28.00'), Decimal('57.00'))
(13L, datetime.datetime(2017, 12, 26, 11, 22, 20), Decimal('28.00'), Decimal('56.00'))
(14L, datetime.datetime(2017, 12, 26, 11, 22, 33), Decimal('28.00'), Decimal('56.00'))
(15L, datetime.datetime(2017, 12, 26, 11, 22, 47), Decimal('28.00'), Decimal('56.00'))
(16L, datetime.datetime(2017, 12, 26, 11, 23), Decimal('28.00'), Decimal('56.00'))
(17L, datetime.datetime(2017, 12, 26, 11, 23, 13), Decimal('28.00'), Decimal('56.00'))
(18L, datetime.datetime(2017, 12, 26, 11, 23, 27), Decimal('28.00'), Decimal('56.00'))
(19L, datetime.datetime(2017, 12, 26, 11, 23, 40), Decimal('28.00'), Decimal('56.00'))
(20L, datetime.datetime(2017, 12, 26, 11, 23, 54), Decimal('28.00'), Decimal('56.00'))
```

Gambar 4. 2 Hasil Pembacaan *Database* Melalui Program Python

Berdasarkan gambar 4.2 dapat disimpulkan bahwa pengujian pengolahan *database* MySQL berhasil.

4.2.4 Pengujian Menampilkan Data pada Grafik

Tujuan pengujian ini yaitu untuk mengetahui apakah data yang disajikan pada grafik sesuai dengan data yang tersimpan pada *database*. Grafik tiga *node* ditampilkan dalam satu laman *website*. Pengujian menampilkan grafik ditunjukkan dalam gambar 4.3.



Gambar 4. 3 Grafik Monitoring Smart Grid

Nilai yang tampil pada grafik sesuai dengan nilai yang terdapat pada *database* MySQL. Namun waktu yang ditunjukkan berbeda dengan waktu yang tersimpan pada *database*.

4.2.5 Pengujian Pengiriman *Command* dari Media Sosial Telegram

Pengujian pengiriman *command* dari dua *user* media sosial Telegram ini adalah untuk mengetahui apakah pesan yang dikirim *user* melalui media sosial Telegram dapat diterima oleh program Python atau tidak. Pengujian dilakukan dengan mengetik pesan melalui media sosial Telegram dan mengirim ke *server* yang dijalankan oleh program Python.

Selain itu, pengujian ini juga bertujuan untuk mengetahui apakah *file* (*.csv) dapat dikirim oleh *server* kepada *user* atau tidak. Pengujian ini dilakukan dengan *user* mengirimkan pesan dengan format tertentu yang memuat informasi tanggal dan waktu. Sehingga program Python akan mengeksekusi fungsi untuk membaca data pada *database* sesuai dengan rentang waktu yang diinginkan.

Tabel 4. 5
Hasil Pengujian Pengiriman *Command*

Pesan Terkirim	Pesan Diterima	Respon Server Python
Name	Name	Smart Grid
Time	Time	2017-12-26 23:34:48.038530
Node1 ON	Node1 ON	Success! Node1 ON
GET1@2017-12-26-11:19:09/ 2017-12-26 16:15:20	GET1@2017-12-26-11:19:09/ 2017-12-26 16:15:20	Report_Node1_2017-12-26-11:19:09/ 2017-12-26 16:15:20
GET2@2017-12-26-11:25:42/ 2017-12-26 11:30:06	GET2@2017-12-26-11:25:42/ 2017-12-26 11:30:06	Report_Node2_2017-12-26-11:25:42/ 2017-12-26 11:30:06
Node3 ON	Node3 ON	Success! Node3 ON
Node1 OFF	Node1 OFF	Success! Node1 OFF
Time	Time	2017-12-26 22:38:17.963710

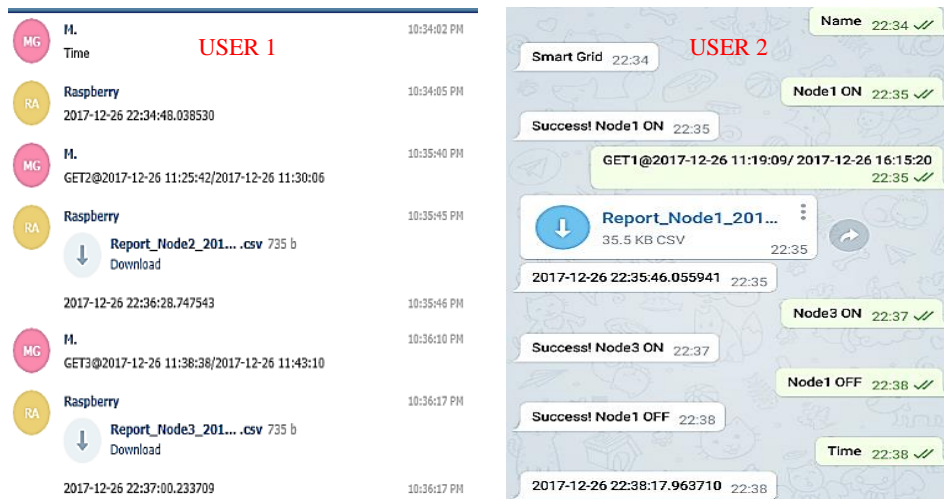
Berdasarkan tabel 4.5 dapat disimpulkan bahwa proses pengiriman pesan dari media sosial Telegram dapat berjalan dengan baik. Semua pesan dapat diterima sesuai dengan yang dikirim oleh *user*.

```

Ready for user1 ...
Ready for user2 ...
User2 Got Command: Name
User1 Got Command : Time
User2 Got Command: Node1 ON
User2 Got Command: GET1@2017-12-26 11:19:09/ 2017-12-26 16:15:20
(1L, datetime.datetime(2017, 12, 26, 11, 19, 9), Decimal('28.00'), Decimal('58.00'))
(2L, datetime.datetime(2017, 12, 26, 11, 19, 22), Decimal('28.00'), Decimal('58.00'))
(3L, datetime.datetime(2017, 12, 26, 11, 19, 40), Decimal('28.00'), Decimal('58.00'))
(4L, datetime.datetime(2017, 12, 26, 11, 19, 54), Decimal('28.00'), Decimal('58.00'))
(5L, datetime.datetime(2017, 12, 26, 11, 20, 16), Decimal('28.00'), Decimal('57.00'))
User1 Got Command : GET2@2017-12-26 11:25:42/2017-12-26 11:30:06
(1L, datetime.datetime(2017, 12, 26, 11, 25, 42), Decimal('28.00'), Decimal('55.00'))
(2L, datetime.datetime(2017, 12, 26, 11, 25, 55), Decimal('28.00'), Decimal('56.00'))
(3L, datetime.datetime(2017, 12, 26, 11, 26, 9), Decimal('28.00'), Decimal('55.00'))
(4L, datetime.datetime(2017, 12, 26, 11, 26, 22), Decimal('28.00'), Decimal('55.00'))
(5L, datetime.datetime(2017, 12, 26, 11, 26, 36), Decimal('28.00'), Decimal('55.00'))
User2 Got Command: Node3 ON
User2 Got Command: Node1 OFF
User2 Got Command: Time

```

Gambar 4. 4 Jendela Program Python Shell



Gambar 4. 5 Screenshot Pengiriman Pesan dari Dua User Media Sosial Telegram

Pesan yang dikirim oleh user akan mendapatkan tanggapan dari *server* apabila pesan tersebut didefinisikan pada program Python, apabila pesan yang dikirim tidak terdapat pada definisi program maka *server* tidak akan memberikan tanggapan atau balasan.

4.2.6 Pengujian ukuran *file* *.csv dan lama waktu pengiriman

Pengujian dilakukan untuk mengamati lama pengiriman data monitoring berdasarkan ukuran data yang berbeda-beda.

Tabel 4. 6

Ukuran Memori *File* *.csv dan Lama Waktu Pengiriman

Row	Memory (KB)	Time (s)
100	3,5	2,91
200	7,1	2,51
300	10,8	3,12
400	14,4	2,64
500	18,0	3,78
600	21,6	4,22
700	25,2	3,39
800	28,8	5,81
900	32,4	3,35
1000	35,5	3,53

Berdasarkan tabel 4.6, didapatkan hasil pengujian pengiriman *file* *.csv kepada *user* memiliki ukuran *file* terbesar yaitu 35.5 KB dengan lama waktu pengiriman 3.53 detik. Sedangkan untuk *file* ukuran 3,5 KB dibutuhkan waktu pengiriman 2.91 detik. Hal ini sebanding dengan banyaknya data yang dibaca oleh *server* dan disimpan dalam *file* tersebut.

4.2.7 Penampil *file* *.csv

Hasil pengiriman *file* *.csv kepada *user* dapat dibuka dengan aplikasi android seperti Excel.

	A	B	C	D
1	ID	Waktu	Tegangan Arus	
2	1	12/26/2017 11:19	28	58
3	2	12/26/2017 11:19	28	58
4	3	12/26/2017 11:19	28	58
5	4	12/26/2017 11:19	28	58
6	5	12/26/2017 11:20	28	57
7	6	12/26/2017 11:20	28	58
8	7	12/26/2017 11:20	28	57
9	8	12/26/2017 11:20	28	57
10	9	12/26/2017 11:21	28	57
11	10	12/26/2017 11:21	28	56
12	11	12/26/2017 11:21	28	56
13	12	12/26/2017 11:21	28	57
14	13	12/26/2017 11:22	28	56
15	14	12/26/2017 11:22	28	56
16	15	12/26/2017 11:22	28	56
17	16	12/26/2017 11:23	28	56
18	17	12/26/2017 11:23	28	56
19	18	12/26/2017 11:23	28	56
20	19	12/26/2017 11:23	28	56
21	20	12/26/2017 11:23	28	56
22				
23				
24				
25				

Gambar 4. 6 Aplikasi Android Penampil File *.csv

4.2.8 Pengujian Pengiriman Data Google Drive

Pengujian ini bertujuan untuk mengetahui keberhasilan pengiriman serta kapasitas (ukuran) data yang tersimpan. Pengujian dilakukan dengan mengirim data dari 100 hingga 500 baris yang memuat informasi nilai tegangan, arus dan waktu ketika data tersebut disimpan pada *database*.

```
pi@raspberrypi:~/Data $ time gdrive-rpi upload Report_Node1_2017-12-26_11:19:09_2017-12-26_15:03:52.csv
Uploading Report_Node1_2017-12-26_11:19:09_2017-12-26_15:03:52.csv
Uploaded 100sJXqz110v4AP6Qe-sqPvq0Vmp0f5 at 1.6 KB/s, total 3.6 KB

real    0m2.253s
user    0m0.310s
sys     0m0.020s
pi@raspberrypi:~/Data $ time gdrive-rpi upload Report_Node1_2017-12-26_11:19:09_2017-12-26_15:12:20Telegram.csv
Uploading Report_Node1_2017-12-26_11:19:09_2017-12-26_15:12:20Telegram.csv
Uploaded 1KkEh1XznRH1j9o0I-Yiv10Dmtejm410z at 4.3 KB/s, total 7.3 KB






real    0m1.795s
user    0m0.330s
sys     0m0.050s
pi@raspberrypi:~/Data $ time gdrive-rpi upload Report_Node1_2017-12-26_11:19:09_2017-12-26_15:20:49Telegram.csv
Uploading Report_Node1_2017-12-26_11:19:09_2017-12-26_15:20:49Telegram.csv
Uploaded 1RjgSTeZqECqrXXY68gdodLWkqFzrgvj at 4.0 KB/s, total 11.0 KB

real    0m2.872s
user    0m0.490s
sys     0m0.040s
pi@raspberrypi:~/Data $ time gdrive-rpi upload Report_Node1_2017-12-26_11:19:09_2017-12-26_15:29:49Telegram.csv
Uploading Report_Node1_2017-12-26_11:19:09_2017-12-26_15:29:49Telegram.csv
Uploaded 11ZiUR1Gwv7bH5kexfQm0vqdv7e6sddo at 1.6 KB/s, total 14.7 KB

real    0m8.997s
user    0m0.490s
sys     0m0.020s
pi@raspberrypi:~/Data $ time gdrive-rpi upload Report_Node1_2017-12-26_11:19:09_2017-12-26_15:38:39Telegram.csv
Uploading Report_Node1_2017-12-26_11:19:09_2017-12-26_15:38:39Telegram.csv
Uploaded 1qP2VqU3e124jk991G1W-MvRELQn04Hq at 11.5 KB/s, total 18.4 KB

real    0m1.687s
user    0m0.290s
sys     0m0.050s
```

Gambar 4. 7 Pengiriman Data pada Server

	Report_Node1_2017-12-26_11:19:09_2017-12-26_15:38:39Telegra...	12.13 Telah diupload	saya	18 KB
	Report_Node1_2017-12-26_11:19:09_2017-12-26_15:29:49Telegra...	12.12 Telah diupload	saya	14 KB
	Report_Node1_2017-12-26_11:19:09_2017-12-26_15:20:49Telegra...	12.11 Telah diupload	saya	11 KB
	Report_Node1_2017-12-26_11:19:09_2017-12-26_15:12:20Telegra...	12.10 Telah diupload	saya	7 KB
	Report_Node1_2017-12-26_11:19:09_2017-12-26_15:03:52.csv	12.09 Telah diupload	saya	4 KB

Gambar 4. 8 Data yang Tersimpan pada Google Drive

Tabel 4. 7
Pengiriman database pada google drive

Row (Database)	Waktu (s)	Memory
100	2,583	4 KB
200	2,175	7 KB
300	3,392	11 KB
400	9,517	14 KB
500	2,027	18 KB

Berdasarkan tabel 4.7 disimpulkan bahwa lama pengiriman data tidak begitu dipengaruhi oleh ukuran data dikarenakan ukuran data yang masih berkisar KB yang kecil, jadi dalam hal ini hanya berpengaruh dari kestabilan koneksi internet yang digunakan.

4.2.9 Pengujian Dua User

Pengujian dilakukan untuk mengetahui selisih waktu yang dibutuhkan masing-masing *user* saat melakukan permintaan data pada waktu yang bersamaan.

Tabel 4. 8

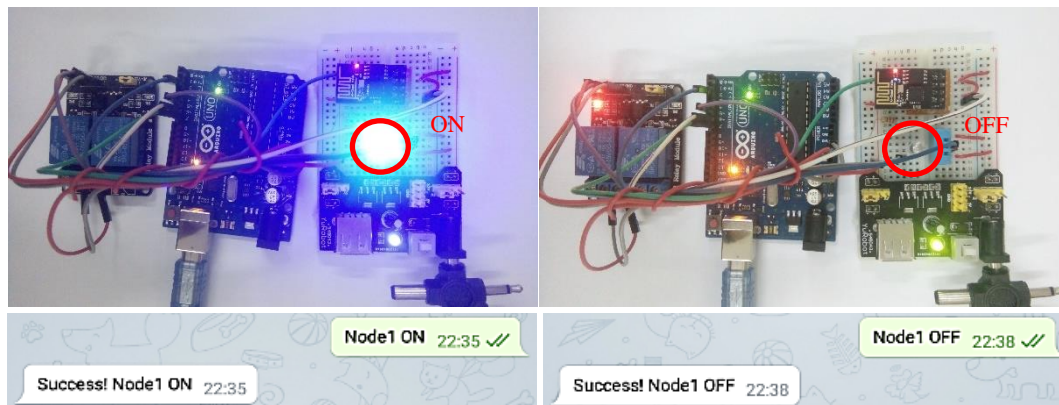
Delay Pengiriman Data Dua User

Data .csv	User 1	User 2
Node 1	3,03 s	2,92 s
	2,66 s	2,97 s
Node 2	3,25 s	4,52 s
	1,62 s	2,88 s
Node 3	3,15 s	3,63 s
	3,07 s	3,16 s
Rata-Rata	2,79 s	3,35 s

Berdasarkan tabel 6 permintaan secara bersamaan pada dua *user* dapat disimpulkan bahwa *user* 1 mengalami proses pengiriman yang lebih cepat dibandingkan *user* 2, dengan selisih waktu pengiriman 0,56 detik.

4.2.10 Pengujian Switching

Pengujian bertujuan mengetahui keberhasilan mengirimkan perintah dari *command switching* pada Telegram. *User* dapat mengirimkan pesan *ON* atau *OFF* pada *node* yang dituju. Proses *switching* pada *smart grid* bertujuan untuk menyambungkan atau memutuskan aliran listrik dari sumber, hal tersebut dibutuhkan untuk penanganan cepat saat terjadi kondisi darurat. Pesan Telegram akan diproses oleh program Python dan dikirimkan ke *node* yang dituju untuk menjalankan perintah. Dalam pengujian ini hanya menggunakan modul relay. Relay sebagai pemutus dan penghubung arus listrik. Sumber tegangan 3,3V digunakan sebagai perumpamaan sebuah sumber listrik pada *grid*, dan LED digunakan sebagai perumpamaan sebuah beban yaitu pengguna listrik pada *grid*. LED yang diamati sesuai dengan kondisi beban *grid* yang terhubung atau terputus dengan sumber listrik. Pengujian *command switching user* yang berhasil direspon relay ditunjukkan oleh gambar 4.9.



Gambar 4. 9 ON/OFF Sistem Sesuai *Command User*

4.3 Pengujian Keseluruhan

Pengujian keseluruhan dilakukan untuk mengetahui unjuk kerja sistem yang telah dirancang. Berikut adalah tabel pengujian keseluruhan sistem yang dilakukan sebanyak 10 kali pengujian.

Tabel 4. 9

Pengujian keseluruhan

Pengujian ke-	Pengiriman data ke Database	Pesan Telegram	Pengiriman File *.csv	Grafik	Google Drive
1	√	√	√	√	√
2	√	√	√	√	√
3	√	√	√	√	√
4	√	√	√	√	√
5	√	√	√	√	√
6	√	√	√	√	√
7	√	√	√	√	√
8	√	√	√	√	√
9	√	√	√	√	√
10	√	√	√	√	√
Error	0 %	0 %	0 %	0 %	0 %

Pada tabel 4.9 menunjukkan bahwa semua proses sistem berjalan dengan baik. Dari proses pengiriman data ke *database* hingga disimpan pada Google Drive berhasil dilakukan dengan error 0%.

BAB V

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Berdasarkan perencanaan, pembuatan dan pengujian yang telah dilakukan terhadap alat baik pengujian pada sub-sistem maupun pengujian seluruh sistem, maka dapat disusun kesimpulan sebagai berikut:

1. Perancangan sistem *switching* dan *monitoring smart grid* jarak jauh berbasis internet menggunakan sebuah *server* Raspberry PI 3 yang berfungsi untuk menerima dan mengolah data pembacaan sensor - sensor yang terpasang pada setiap *node*, sehingga data dapat dikirim ke *user* melalui aplikasi media sosial Telegram. Data dari sensor akan diolah oleh *script* PHP untuk disimpan pada *database* MySQL. Program Python akan mengirimkan *file monitoring* sesuai permintaan *user* yang kemudian dapat dibuka dengan aplikasi android seperti Excel serta dapat memproses perintah *ON* atau *OFF* sistem dari *user* untuk *node* yang dituju.
2. Data tegangan dan arus yang diterima dari modul sensor diolah oleh *script* PHP dan disimpan dalam tabel masing-masing *node* pada *database* MySQL. Waktu yang dibutuhkan pengiriman secara *wireless* adalah 14,33 detik dengan *delay* pembacaan sensor 10 detik. Data yang tersimpan dapat ditampilkan dalam bentuk grafik *monitoring* dalam satu laman *website*. Namun untuk waktu yang tampil pada grafik tidak sinkron dengan waktu yang tersimpan pada *database*. Program Python mengolah pengiriman data ke Telegram berupa file *.csv serta dapat menyimpan data pada Google Drive.
3. Proses *switching* pada *smart grid* bertujuan untuk menyambungkan atau memutuskan aliran listrik dari sumber untuk penanganan cepat saat terjadi kondisi darurat. Proses *switching* dilakukan dengan mengirimkan pesan pada Telegram berupa perintah *ON* atau *OFF* untuk *node* yang dituju. Python memproses pesan yang diterima kemudian mengirimkan sebuah perintah pada *node* yang dituju.

5.2 Saran

Saran yang dapat digunakan dalam meningkatkan kerja sistem adalah penambahan fitur untuk tampilan *web* yang baik pada *grafik smart grid*. Sehingga tampilan menjadi lebih menarik dan tepat dalam menampilkan data waktu. Serta dapat dikembangkan menggunakan aplikasi android tersendiri sebagai user.

DAFTAR PUSTAKA

- Anisya. (2013). Aplikasi Sistem *Database* Rumah Sakit Terpusat pada Rumah Sakit Umum (RSU) ‘Aisyiyah Padang dengan Menerapkan *Open Source* (PHP – MySQL). *Jurnal Momentum*. XV (2): 49-58.
- Arianes, J. (2016). Sistem Pemantauan Ruangan Jarak Jauh Via Internet Berbasis Raspberry Pi 2 Terintegrasi dengan Google Drive dan Line. *Jurnal Universitas Brawijaya, Malang*.
- Binanto, I. (2005). Membangun *Web Server* (Apache+PHP+MySQL) menggunakan FreeBSD.
- BPPT Press. (2012). Teknologi *Smart Grid* untuk *Smart City*. Jakarta: BPPT
- Ditjen Ketenagalistrikan. (2016). *Statistik Ketenagalistrikan 2015*. Jakarta: Ditjen Ketenagalistrikan
- Downey, A. (2012). *Think Python*. USA: Creative Commons
- Firmansyah, M. (2017). Sistem Monitoring Tegangan, Arus, dan Daya pada *Smart Grid* Berbasis Raspberry Pi 3 Terintegrasi dengan Media Sosial Telegram dan *Website*. *Skripsi*. Tidak dipublikasi. Malang: Universitas Brawijaya.
- Hakim, M. A. I. & Putra, Y. H. (2013). Pemanfaatan Mini PC Raspberry Pi Sebagai Pengontrol Jarak Jauh Berbasis *Web* pada Rumah. *Jurnal Jurusan Teknik Komputer Unikom, Bandung*. 1-6.
- Iman, M. C. (2015). Rancang Bangun Transmisi Data Berbasis *Wireless* pada *Single Smart Grid*. *Skripsi*. Tidak dipublikasi. Malang: Universitas Brawijaya.
- Kholiq, I. (2015). Pemanfaatan Energi Alternatif Sebagai Energi Terbarukan untuk Mendukung Substitusi BBM. *Jurnal IPTEK*. XIX (2): 75-91.
- Perkasa, T. R. & Widyantara, H. & Susanto, P. (2014). Rancang Bangun Pendeteksi Gerak Menggunakan Metode *Image Subtraction* pada *Single Board Computer* (SBC). *Journal of Control and Network Systems*. III (2): 90-97.
- Ramadhani, G. (2003). Modul Pengenalan Internet. Malang: Universitas Muhammadiyah Malang.
- Saputra, A. (2012). Manajemen Basis Data MySQL pada Situs FTP Lapan Bandung. *Jurnal Berita Dirgantara*. XIII (4): 155-162.
- Sirait, F. (2015). Sistem *Monitoring* Keamanan Gedung berbasis Raspberry Pi. *Jurnal Teknologi Elektro, Universitas Mercu Buana*. VI (1): 55-60.
- Sulastri. (2008). Rekayasa Perangkat Lunak *Database* Jurnal Ilmiah Berbasis *Web* Menggunakan PHP dan MySQL. *Jurnal MMI Fakultas Kedokteran UniversitasDiponegoro*. XIII (1): 73-79.